Signals and Systems

with MATLAB ® Computing and Simulink ® Modeling

Fourth Edition

Steven T. Karris

$$X[m] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{mn}{N}}$$

Includes

step-by-step

procedures

for designing

analog and

digital filters

Orchard Publications
www.orchardpublications.com

Signals and Systems
with MATLAB ® Computing and Simulink ® Modeling
Fourth Edition

Students and working professionals will find Signals and Systems with MATLAB ® Computing and Simulink ® Modeling, Fourth Edition, to be a concise and easy-to-learn text. It provides complete, clear, and detailed explanations of the principal analog and digital signal processing concepts and analog and digital filter design illustrated with numerous practical examples.

This text includes the following chapters and appendices:

• Elementary Signals • The Laplace Transformation • The Inverse Laplace Transformation • Circuit Analysis with Laplace Transforms • State Variables and State Equations • The Impulse Response and Convolution • Fourier Series • The Fourier Transform • Discrete Time Systems and the Z Transform • The DFT and The FFT Algorithm • Analog and Digital Filters • Introduction to MATLAB ® • Introduction to Simulink ® • Review of Complex Numbers • Review of Matrices and Determinants • Window Functions

Each chapter contains numerous practical applications supplemented with detailed instructions for using MATLAB and Simulink to obtain accurate and quick solutions.

Steven T. Karris is the founder and president of Orchard Publications, has undergraduate and graduate degrees in electrical engineering, and is a registered professional engineer in California and Florida. He has more than 35 years of professional engineering experience and more than 30 years of teaching experience as an adjunct professor, most recently at UC Berkeley, California. His area of interest is in The MathWorks, Inc. ™ products and the publication of MATLAB® and Simulink® based texts.

Orchard Publications
Visit us on the Internet
www.orchardpublications.comor email us:
info@orchardpublications.com

ISBN 978-1-934404-11-9
57995
9 781934 404119

ISBN-10: 1-934404-11-X

ISBN-13: 978-1-934404-11-9

$79.95 U.S.A.

Preface

This text contains a comprehensive discussion on continuous and discrete time signals and systems with many MATLAB® and several Simulink® examples. It is written for junior and senior electrical and computer engineering students, and for self–study by working professionals. The prerequisites are a basic course in differential and integral calculus, and basic electric circuit theory.

This book can be used in a two–quarter, or one semester course. This author has taught the subject material for many years and was able to cover all material in 16 weeks, with 2½ lecture hours per week.

To get the most out of this text, it is highly recommended that Appendix A is thoroughly reviewed. This appendix serves as an introduction to MATLAB, and is intended for those who are not familiar with it. The Student Edition of MATLAB is an inexpensive, and yet a very powerful software package; it can be found in many college bookstores, or can be obtained directly from

The MathWorks™ Inc., 3 Apple Hill Drive, Natick, MA 01760–2098
Phone: 508 647–7000, Fax: 508 647–7001
http://www.mathworks.com
e–mail: info@mathworks.com

The elementary signals are reviewed in Chapter 1, and several examples are given. The purpose of this chapter is to enable the reader to express any waveform in terms of the unit step function, and subsequently the derivation of the Laplace transform of it. Chapters 2 through 4 are devoted to Laplace transformation and circuit analysis using this transform. Chapter 5 is an introduction to state–space and contains many illustrative examples. Chapter 6 discusses the impulse response. Chapters 7 and 8 are devoted to Fourier series and transform respectively. Chapter 9 introduces discrete–time signals and the $\mathcal{Z}$ transform. Considerable time was spent on Chapter 10 to present the Discrete Fourier transform and FFT with the simplest possible explanations. Chapter 11 contains a thorough discussion to analog and digital filters analysis and design procedures. As mentioned above, Appendix A is an introduction to MATLAB. Appendix B is an introduction to Simulink, Appendix C contains a review of complex numbers, and Appendix D is an introduction to matrix theory.

New to the Second Edition

This is an extensive revision of the first edition. The most notable change is the inclusion of the solutions to all exercises at the end of each chapter. It is in response to many readers who expressed a desire to obtain the solutions in order to check their solutions to those of the author and thereby enhancing their knowledge. Another reason is that this text is written also for self–

study by practicing engineers who need a review before taking more advanced courses such as digital image processing.

Another major change is the addition of a rather comprehensive summary at the end of each chapter. Hopefully, this will be a valuable aid to instructors for preparation of view foils for presenting the material to their class.

New to the Third Edition

The most notable change is the inclusion of Simulink modeling examples. The pages where they appear can be found in the Table of Contents section of this text. Another change is the improvement of the plots generated by the latest revisions of the MATLAB® Student Version, Release 14.

The author wishes to express his gratitude to the staff of The MathWorks™, the developers of MATLAB® and Simulink®, especially to Ms. Courtney Esposito, for the encouragement and unlimited support they have provided me with during the production of this text.

Our heartfelt thanks also to Ms. Sally Wright, P.E., of Renewable Energy Research Laboratory University of Massachusetts, Amherst, for bringing some errors on the previous editions to our attention.

New to the Fourth Edition

The most notable change is the inclusion of Appendix E on window functions. The plots were generated generated with the latest revisions of the MATLAB® R2008a edition. Also, two end-of- chapter exercises were added in Chapter 10 to illustrate the use of the **fft** and **ifft** MATLAB functions

The author wishes to express his gratitude to the staff of The MathWorks™, the developers of MATLAB® and Simulink®, especially to The MathWorks™ Book Program Team, for the encouragement and unlimited support they have provided me with during the production of this and all other texts by this publisher.

Orchard Publications
www.orchardpublications.com
info@orchardpublications.com

Table of Contents

1 *Elementary Signals* 1–1

1.1 Signals Described in Math Form 1–1
1.2 The Unit Step Function 1–2
1.3 The Unit Ramp Function 1–10
1.4 The Delta Function 1–11
  1.4.1 The Sampling Property of the Delta Function 1–12
  1.4.2 The Sifting Property of the Delta Function 1–13
1.5 Higher Order Delta Functions 1–14
1.6 Summary 1–22
1.7 Exercises 1–23
1.8 Solutions to End–of–Chapter Exercises 1–24

MATLAB Computing
Pages 1–20, 1–21

Simulink Modeling
Page 1–18

2 *The Laplace Transformation* 2–1

2.1 Definition of the Laplace Transformation 2–1
2.2 Properties and Theorems of the Laplace Transform 2–2
  2.2.1 Linearity Property 2–3
  2.2.2 Time Shifting Property 2–3
  2.2.3 Frequency Shifting Property 2–4
  2.2.4 Scaling Property 2–4
  2.2.5 Differentiation in Time Domain Property 2–4
  2.2.6 Differentiation in Complex Frequency Domain Property 2–6
  2.2.7 Integration in Time Domain Property 2–6
  2.2.8 Integration in Complex Frequency Domain Property 2–8
  2.2.9 Time Periodicity Property 2–8
  2.2.10 Initial Value Theorem 2–9
  2.2.11 Final Value Theorem 2–10
  2.2.12 Convolution in Time Domain Property 2–11
  2.2.13 Convolution in Complex Frequency Domain Property 2–12
2.3 The Laplace Transform of Common Functions of Time 2–14
  2.3.1 The Laplace Transform of the Unit Step Function $u_0(t)$ 2–14
  2.3.2 The Laplace Transform of the Ramp Function $u_1(t)$ 2–14
  2.3.3 The Laplace Transform of $t^n u_0(t)$ 2–15

2.3.4 The Laplace Transform of the Delta Function $\delta(t)$ 2–18
2.3.5 The Laplace Transform of the Delayed Delta Function $\delta(t-a)$ 2–18
2.3.6 The Laplace Transform of $e^{-at}u_0(t)$ 2–19
2.3.7 The Laplace Transform of $t^n e^{-at}u_0(t)$ 2–19
2.3.8 The Laplace Transform of $\sin\omega t\ u_0 t$ 2–20
2.3.9 The Laplace Transform of $\cos\omega t\ u_0 t$ 2–20
2.3.10 The Laplace Transform of $e^{-at}\sin\omega t\ u_0(t)$ 2–21
2.3.11 The Laplace Transform of $e^{-at}\cos\omega t\ u_0(t)$ 2–22
2.4 The Laplace Transform of Common Waveforms 2–23
2.4.1 The Laplace Transform of a Pulse 2–23
2.4.2 The Laplace Transform of a Linear Segment 2–23
2.4.3 The Laplace Transform of a Triangular Waveform 2–24
2.4.4 The Laplace Transform of a Rectangular Periodic Waveform 2–25
2.4.5 The Laplace Transform of a Half–Rectified Sine Waveform 2–26
2.5 Using MATLAB for Finding the Laplace Transforms of Time Functions 2–27
2.6 Summary 2–28
2.7 Exercises 2–31
The Laplace Transform of a Sawtooth Periodic Waveform 2–32
The Laplace Transform of a Full–Rectified Sine Waveform 2–32
2.8 Solutions to End–of–Chapter Exercises 2–33

3 *The Inverse Laplace Transform* 3–1

3.1 The Inverse Laplace Transform Integral 3–1
3.2 Partial Fraction Expansion 3–1
3.2.1 Distinct Poles 3–2
3.2.2 Complex Poles 3–5
3.2.3 Multiple (Repeated) Poles 3–8
3.3 Case where F(s) is Improper Rational Function 3–13
3.4 Alternate Method of Partial Fraction Expansion 3–15
3.5 Summary 3–19
3.6 Exercises 3–21
3.7 Solutions to End–of–Chapter Exercises 3–22

MATLAB Computing
Pages 3–3, 3–4, 3–5, 3–6, 3–8, 3–10, 3–12, 3–13, 3–14, 3–22

4 *Circuit Analysis with Laplace Transforms* 4–1

4.1 Circuit Transformation from Time to Complex Frequency 4–1
4.1.1 Resistive Network Transformation 4–1
4.1.2 Inductive Network Transformation 4–1
4.1.3 Capacitive Network Transformation 4–1

4.2 Complex Impedance Z(s) 4–8
4.3 Complex Admittance Y(s) 4–11
4.4 Transfer Functions 4–13
4.5 Using the Simulink Transfer Fcn Block 4–17
4.6 Summary 4–20
4.7 Exercises 4–21
4.8 Solutions to End–of–Chapter Exercises 4–24

MATLAB Computing
Pages 4–6, 4–8, 4–12, 4–16, 4–17, 4–18, 4–26, 4–27, 4–28, 4–29, 4–34

Simulink Modeling
Page 4–17

5 ***State Variables and State Equations*** 5–1

5.1 Expressing Differential Equations in State Equation Form 5–1
5.2 Solution of Single State Equations 5–6
5.3 The State Transition Matrix 5–9
5.4 Computation of the State Transition Matrix 5–11
  5.4.1 Distinct Eigenvalues 5–11
  5.4.2 Multiple (Repeated) Eigenvalues 5–15
5.5 Eigenvectors 5–18
5.6 Circuit Analysis with State Variables 5–22
5.7 Relationship between State Equations and Laplace Transform 5–30
5.8 Summary 5–38
5.9 Exercises 5–41
5.10 Solutions to End–of–Chapter Exercises 5–43

MATLAB Computing
Pages 5–14, 5–15, 5–18, 5–26, 5–36, 5–48, 5–51

Simulink Modeling
Pages 5–27, 5–37, 5–45

6 ***The Impulse Response and Convolution*** 6–1

6.1 The Impulse Response in Time Domain 6–1
6.2 Even and Odd Functions of Time 6–4
6.3 Convolution 6–7
6.4 Graphical Evaluation of the Convolution Integral 6–8
6.5 Circuit Analysis with the Convolution Integral 6–18
6.6 Summary 6–21
6.7 Exercises 6–23

6.8 Solutions to End–of–Chapter Exercises 6–25

MATLAB Applications
Pages 6–12, 6–15, 6–30

7 *Fourier Series* 7–1

7.1 Wave Analysis 7–1
7.2 Evaluation of the Coefficients 7–2
7.3 Symmetry in Trigonometric Fourier Series 7–6
    7.3.1 Symmetry in Square Waveform 7–8
    7.3.2 Symmetry in Square Waveform with Ordinate Axis Shifted 7–8
    7.3.3 Symmetry in Sawtooth Waveform 7–9
    7.3.4 Symmetry in Triangular Waveform 7–9
    7.3.5 Symmetry in Fundamental, Second, and Third Harmonics 7–10
7.4 Trigonometric Form of Fourier Series for Common Waveforms 7–10
    7.4.1 Trigonometric Fourier Series for Square Waveform 7–11
    7.4.2 Trigonometric Fourier Series for Sawtooth Waveform 7–14
    7.4.3 Trigonometric Fourier Series for Triangular Waveform 7–16
    7.4.4 Trigonometric Fourier Series for Half–Wave Rectifier Waveform 7–17
    7.4.5 Trigonometric Fourier Series for Full–Wave Rectifier Waveform 7–20
7.5 Gibbs Phenomenon 7–24
7.6 Alternate Forms of the Trigonometric Fourier Series 7–24
7.7 Circuit Analysis with Trigonometric Fourier Series 7–28
7.8 The Exponential Form of the Fourier Series 7–31
7.9 Symmetry in Exponential Fourier Series 7–33
    7.9.1 Even Functions 7–33
    7.9.2 Odd Functions 7–34
    7.9.3 Half-Wave Symmetry 7–34
    7.9.4 No Symmetry 7–34
    7.9.5 Relation of C_{-n} to C_n 7–34
7.10 Line Spectra 7–36
7.11 Computation of RMS Values from Fourier Series 7–41
7.12 Computation of Average Power from Fourier Series 7–44
7.13 Evaluation of Fourier Coefficients Using Excel® 7–46
7.14 Evaluation of Fourier Coefficients Using MATLAB® 7–47
7.15 Summary 7–50
7.16 Exercises 7–53
7.17 Solutions to End–of–Chapter Exercises 7–55

MATLAB Computing
Pages 7–38, 7–47

Simulink Modeling
Page 7–31

8 *The Fourier Transform* 8–1

8.1 Definition and Special Forms 8–1
8.2 Special Forms of the Fourier Transform 8–2
  8.2.1 Real Time Functions 8–3
  8.2.2 Imaginary Time Functions 8–6
8.3 Properties and Theorems of the Fourier Transform 8–9
  8.3.1 Linearity 8–9
  8.3.2 Symmetry 8–9
  8.3.3 Time Scaling 8–10
  8.3.4 Time Shifting 8–11
  8.3.5 Frequency Shifting 8–11
  8.3.6 Time Differentiation 8–12
  8.3.7 Frequency Differentiation 8–13
  8.3.8 Time Integration 8–13
  8.3.9 Conjugate Time and Frequency Functions 8–13
  8.3.10 Time Convolution 8–14
  8.3.11 Frequency Convolution 8–15
  8.3.12 Area Under $f(t)$ 8–15
  8.3.13 Area Under $F(\omega)$ 8–15
  8.3.14 Parseval's Theorem 8–16
8.4 Fourier Transform Pairs of Common Functions 8–18
  8.4.1 The Delta Function Pair 8–18
  8.4.2 The Constant Function Pair 8–18
  8.4.3 The Cosine Function Pair 8–19
  8.4.4 The Sine Function Pair 8–20
  8.4.5 The Signum Function Pair 8–20
  8.4.6 The Unit Step Function Pair 8–22
  8.4.7 The $e^{-j\omega_0 t}u_0(t)$ Function Pair 8–24
  8.4.8 The $(\cos\omega_0 t)(u_0 t)$ Function Pair 8–24
  8.4.9 The $(\sin\omega_0 t)(u_0 t)$ Function Pair 8–25
8.5 Derivation of the Fourier Transform from the Laplace Transform 8–25
8.6 Fourier Transforms of Common Waveforms 8–27
  8.6.1 The Transform of $f(t) = A[u_0(t+T) - u_0(t-T)]$ 8–27
  8.6.2 The Transform of $f(t) = A[u_0(t) - u_0(t-2T)]$ 8–28
  8.6.3 The Transform of $f(t) = A[u_0(t+T) + u_0(t) - u_0(t-T) - u_0(t-2T)]$ 8–29

8.6.4 The Transform of $f(t) = A\cos\omega_0 t[u_0(t+T) - u_0(t-T)]$ 8–30
8.6.5 The Transform of a Periodic Time Function with Period T 8–31
8.6.6 The Transform of the Periodic Time Function $f(t) = A\sum_{n=-\infty}^{\infty}\delta(t-nT)$ 8–32
8.7 Using MATLAB for Finding the Fourier Transform of Time Functions 8–33
8.8 The System Function and Applications to Circuit Analysis 8–34
8.9 Summary 8–42
8.10 Exercises 8–47
8.11 Solutions to End–of–Chapter Exercises 8–49

MATLAB Computing

Pages 8–33, 8–34, 8–50, 8–54, 8–55, 8–56, 8–59, 8–60

9 *Discrete–Time Systems and the $\mathcal{Z}$ Transform* 9–1

9.1 Definition and Special Forms of the $\mathcal{Z}$ Transform 9–1
9.2 Properties and Theorems of the $\mathcal{Z}$ Transform 9–3
9.2.1 Linearity 9–3
9.2.2 Shift of $f[n]u_0[n]$ in the Discrete–Time Domain 9–3
9.2.3 Right Shift in the Discrete–Time Domain 9–4
9.2.4 Left Shift in the Discrete–Time Domain 9–5
9.2.5 Multiplication by a^n in the Discrete–Time Domain 9–6
9.2.6 Multiplication by e^{-naT} in the Discrete–Time Domain 9–6
9.2.7 Multiplication by n and n^2 in the Discrete–Time Domain 9–6
9.2.8 Summation in the Discrete–Time Domain 9–7
9.2.9 Convolution in the Discrete–Time Domain 9–8
9.2.10 Convolution in the Discrete–Frequency Domain 9–9
9.2.11 Initial Value Theorem 9–9
9.2.12 Final Value Theorem 9–10
9.3 The $\mathcal{Z}$ Transform of Common Discrete–Time Functions 9–11
9.3.1 The Transform of the Geometric Sequence 9–11
9.3.2 The Transform of the Discrete–Time Unit Step Function 9–14
9.3.3 The Transform of the Discrete–Time Exponential Sequence 9–16
9.3.4 The Transform of the Discrete–Time Cosine and Sine Functions 9–16
9.3.5 The Transform of the Discrete–Time Unit Ramp Function 9–18
9.4 Computation of the $\mathcal{Z}$ Transform with Contour Integration 9–20
9.5 Transformation Between s– and z–Domains 9–22
9.6 The Inverse $\mathcal{Z}$ Transform 9–25

9.6.1 Partial Fraction Expansion 9–25
9.6.2 The Inversion Integral 9–32
9.6.3 Long Division of Polynomials 9–36
9.7 The Transfer Function of Discrete–Time Systems 9–38
9.8 State Equations for Discrete–Time Systems 9–45
9.9 Summary 9–48
9.10 Exercises 9–53
9.11 Solutions to End–of–Chapter Exercises 9–55

MATLAB Computing
Pages 9–35, 9–37, 9–38, 9–41, 9–42, 9–59, 9–61

Simulink Modeling
Page 9–44

Excel Plots
Pages 9–35, 9–44

10 *The DFT and the FFT Algorithm* 10–1

10.1 The Discrete Fourier Transform (DFT) 10–1
10.2 Even and Odd Properties of the DFT 10–9
10.3 Common Properties and Theorems of the DFT 10–10
10.3.1 Linearity 10–10
10.3.2 Time Shift 10–11
10.3.3 Frequency Shift 10–12
10.3.4 Time Convolution 10–12
10.3.5 Frequency Convolution 10–13
10.4 The Sampling Theorem 10–13
10.5 Number of Operations Required to Compute the DFT 10–16
10.6 The Fast Fourier Transform (FFT) 10–17
10.7 Summary 10–28
10.8 Exercises 10–31
10.9 Solutions to End–of–Chapter Exercises 10–33

MATLAB Computing
Pages 10–5, 10–7, 10–34

Excel Analysis ToolPak
Pages 10–6, 10–8

11 *Analog and Digital Filters*

11.1 Filter Types and Classifications 11–1
11.2 Basic Analog Filters 11–2

11.2.1 RC Low–Pass Filter 11–2
11.2.2 RC High–Pass Filter 11–4
11.2.3 RLC Band–Pass Filter 11–7
11.2.4 RLC Band–Elimination Filter 11–8
11.3 Low–Pass Analog Filter Prototypes 11–10
11.3.1 Butterworth Analog Low–Pass Filter Design 11–14
11.3.2 Chebyshev Type I Analog Low–Pass Filter Design 11–25
11.3.3 Chebyshev Type II Analog Low–Pass Filter Design 11–38
11.3.4 Elliptic Analog Low–Pass Filter Design 11–39
11.4 High–Pass, Band–Pass, and Band–Elimination Filter Design 11–41
11.5 Digital Filters 11–51
11.6 Digital Filter Design with Simulink 11–70
11.6.1 The Direct Form I Realization of a Digital Filter 11–70
11.6.2 The Direct Form II Realization of a Digital Filter 11–71
11.6.3 The Series Form Realization of a Digital Filter 11–73
11.6.4 The Parallel Form Realization of a Digital Filter 11–75
11.6.5 The Digital Filter Design Block 11–78
11.7 Summary 11–87
11.8 Exercises 11–91
11.9 Solutions to End–of–Chapter Exercises 11–97

MATLAB Computing
Pages 11–3, 11–4, 11–6, 11–7, 11–9, 11–15, 11–19, 11–23, 11–24, 11–31, 11–35, 11–36, 11–37, 11–38, 11–40, 11–41, 11–42, 11–43, 11–45, 11–46, 11–48, 11–50, 11–55, 11–56, 11–57, 11–60, 11–62, 11–64, 11–67, 11–68, and 11–97 through 11–106

Simulink Modeling
Pages 11–71, 11–74, 11–77, 11–78, 11–80, 11–82, 11–83, 11–84

A ***Introduction to MATLAB*** A–1

A.1 MATLAB® and Simulink® A–1
A.2 Command Window A–1
A.3 Roots of Polynomials A–3
A.4 Polynomial Construction from Known Roots A–4
A.5 Evaluation of a Polynomial at Specified Values A–6
A.6 Rational Polynomials A–8
A.7 Using MATLAB to Make Plots A–10
A.8 Subplots A–18
A.9 Multiplication, Division, and Exponentiation A–18
A.10 Script and Function Files A–26
A.11 Display Formats A–31

MATLAB Computing
Pages A–3 through A–8, A–10, A–13, A–14, A–16, A–17, A–21, A–22, A–24, A–27

B

Introduction to Simulink B–1

B.1 Simulink and its Relation to MATLAB B–1
B.2 Simulink Demos B–20

MATLAB Computing
Page B–4

Simulink Modeling
Pages B–7, B–12, B–14, B–18

C

A Review of Complex Numbers C–1

C.1 Definition of a Complex Number C–1
C.2 Addition and Subtraction of Complex Numbers C–2
C.3 Multiplication of Complex Numbers C–3
C.4 Division of Complex Numbers C–4
C.5 Exponential and Polar Forms of Complex Numbers C–4

MATLAB Computing
Pages C–6, C–7, C–8

Simulink Modeling
Page C–7

D

Matrices and Determinants D–1

D.1 Matrix Definition D–1
D.2 Matrix Operations D–2
D.3 Special Forms of Matrices D–6
D.4 Determinants D–10
D.5 Minors and Cofactors D–12
D.6 Cramer’s Rule D–17
D.7 Gaussian Elimination Method D–19
D.8 The Adjoint of a Matrix D–21
D.9 Singular and Non–Singular Matrices D–21
D.10 The Inverse of a Matrix D–22
D.11 Solution of Simultaneous Equations with Matrices D–24
D.12 Exercises D–31

MATLAB Computing
Pages D–3, D–4, D–5, D–7, D–8, D–9, D–10,
D–12, D–19, D–23, D–27, D–29

Simulink Modeling
Page D–3

Excel Spreadsheet
Page D–28

E *Window Functions* E–1

E.1 Window Function Defined E–1
E.2 Common Window Functions E–1
  E.2.1 Rectangular Window Function E–2
  E.2.2 Triangular Window Function E–5
  E.2.3 Hanning Window Function E–7
  E.2.4 Hamming Window Function E–9
  E.2.5 Blackman Window Function E–12
  E.2.6 Kaiser Family of Window Functions E–14
E.3 Other Window Functions E–15
E.4 Fourier Series Method for Approximating an FIR Amplitude Response E–17

References R–1

Index IN1

We can express (1.3) by the waveform shown in Figure 1.2.

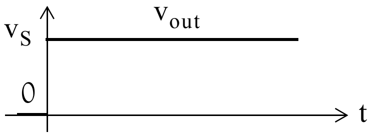


Figure 1.2. Waveform for v_{out} as defined in relation (1.3)

The waveform of Figure 1.2 is an example of a discontinuous function. A function is said to be *discontinuous* if it exhibits points of discontinuity, that is, the function jumps from one value to another without taking on any intermediate values.

1.2 The Unit Step Function $u_0(t)$

A well known discontinuous function is the *unit step function* $u_0(t)$ * which is defined as

$$\boxed{u_0(t) = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases}} \tag{1.4}$$

It is also represented by the waveform of Figure 1.3.

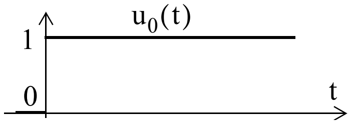


Figure 1.3. Waveform for $u_0(t)$

In the waveform of Figure 1.3, the unit step function $u_0(t)$ changes abruptly from 0 to 1 at $t = 0$. But if it changes at $t = t_0$ instead, it is denoted as $u_0(t - t_0)$. In this case, its waveform and definition are as shown in Figure 1.4 and relation (1.5) respectively.

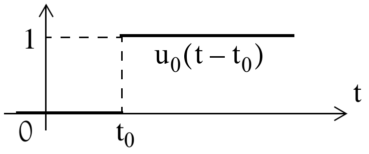


Figure 1.4. Waveform for $u_0(t - t_0)$

---

* *In some books, the unit step function is denoted as* $u(t)$*, that is, without the subscript 0. In this text, however, we will reserve the* $u(t)$ *designation for any input when we will discuss state variables in Chapter 5.*

Thus, the pulse of Figure 1.9(a) is the sum of the unit step functions of Figures 1.9(b) and 1.9(c) and it is represented as $u_0(t) - u_0(t-1)$.

The unit step function offers a convenient method of describing the sudden application of a voltage or current source. For example, a constant voltage source of $24\ \text{V}$ applied at $t = 0$, can be denoted as $24u_0(t)\ \text{V}$. Likewise, a sinusoidal voltage source $v(t) = V_m \cos\omega t\ \text{V}$ that is applied to a circuit at $t = t_0$, can be described as $v(t) = (V_m \cos\omega t) u_0(t - t_0)\ \text{V}$. Also, if the excitation in a circuit is a rectangular, or triangular, or sawtooth, or any other recurring pulse, it can be represented as a sum (difference) of unit step functions.

Example 1.2

Express the square waveform of Figure 1.10 as a sum of unit step functions. The vertical dotted lines indicate the discontinuities at $T, 2T, 3T$, and so on.

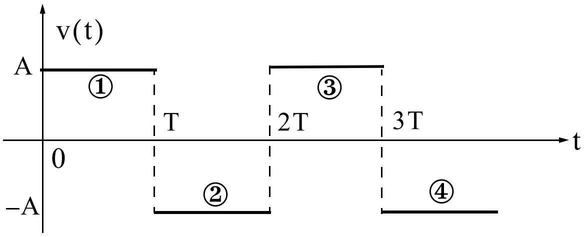


Figure 1.10. Square waveform for Example 1.2

Solution:

Line segment ① has height A, starts at $t = 0$, and terminates at $t = T$. Then, as in Example 1.1, this segment is expressed as

$$v_1(t) = A[u_0(t) - u_0(t-T)] \tag{1.8}$$

Line segment ② has height $-A$, starts at $t = T$ and terminates at $t = 2T$. This segment is expressed as

$$v_2(t) = -A[u_0(t-T) - u_0(t-2T)] \tag{1.9}$$

Line segment ③ has height A, starts at $t = 2T$ and terminates at $t = 3T$. This segment is expressed as

$$v_3(t) = A[u_0(t-2T) - u_0(t-3T)] \tag{1.10}$$

Line segment ④ has height $-A$, starts at $t = 3T$, and terminates at $t = 4T$. It is expressed as

$$v_4(t) = -A[u_0(t-3T) - u_0(t-4T)] \tag{1.11}$$

$$\int_{-\infty}^{t} f(\tau)\delta(\tau)d\tau = \int_{-\infty}^{t} f(0)\delta(\tau)d\tau + \int_{-\infty}^{t} [f(\tau)-f(0)]\delta(\tau)d\tau \tag{1.39}$$

The first integral on the right side of (1.39) contains the constant term $f(0)$; this can be written outside the integral, that is,

$$\int_{-\infty}^{t} f(0)\delta(\tau)d\tau = f(0)\int_{-\infty}^{t} \delta(\tau)d\tau \tag{1.40}$$

The second integral of the right side of (1.39) is always zero because

$$\delta(t) = 0 \ \text{ for } \ t<0 \ \text{ and } \ t>0$$

and

$$[f(\tau)-f(0)]\big|_{\tau=0} = f(0)-f(0) = 0$$

Therefore, (1.39) reduces to

$$\int_{-\infty}^{t} f(\tau)\delta(\tau)d\tau = f(0)\int_{-\infty}^{t} \delta(\tau)d\tau \tag{1.41}$$

Differentiating both sides of (1.41), and replacing τ with t, we obtain

$$\boxed{\begin{array}{c} f(t)\delta(t) = f(0)\delta(t) \\ \text{Sampling Property of } \delta(t) \end{array}} \tag{1.42}$$

1.4.2 The Sifting Property of the Delta Function $\delta(t)$

The *sifting property* of the delta function states that

$$\boxed{\int_{-\infty}^{\infty} f(t)\delta(t-\alpha)dt = f(\alpha)} \tag{1.43}$$

that is, if we multiply any function $f(t)$ by $\delta(t-\alpha)$, and integrate from $-\infty$ to $+\infty$, we will obtain the value of $f(t)$ evaluated at $t = \alpha$.

Proof:

Let us consider the integral

$$\int_{a}^{b} f(t)\delta(t-\alpha)dt \quad \text{where} \quad a<\alpha<b \tag{1.44}$$

We will use integration by parts to evaluate this integral. We recall from the derivative of products that

$$d(xy) = xdy+ydx \quad \text{or} \quad xdy = d(xy)-ydx \tag{1.45}$$

and integrating both sides we obtain

6. To drag a line segment to a new position, we place the mouse cursor over that line segment and the cursor shape shows the position in which we can drag the segment.
7. To drag a point along the y−axis, we move the mouse cursor over that point, and the cursor changes to a circle indicating that we can drag that point. Then, we can move that point in a direction parallel to the x−axis.
8. To drag a point along the x−axis, we select that point, and we hold down the Shift key while dragging that point.
9. When we select a line segment on the time axis (x−axis) we observe that at the lower end of the waveform display window the **Left Point** and **Right Point** fields become visible. We can then reshape the given waveform by specifying the Time (**T**) and Amplitude (**Y**) points.

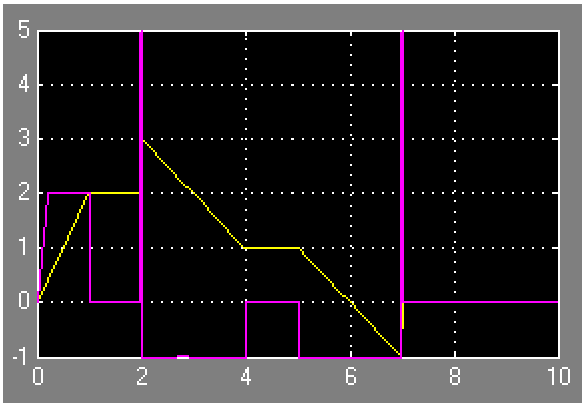


Figure 1.26. Waveforms for the Simulink model of Figure 1.24

The two positive spikes that occur at $t = 2$, and $t = 7$, are clearly shown in Figure 1.26. MATLAB* has built-in functions for the unit step, and the delta functions. These are denoted by the names of the mathematicians who used them in their work. The unit step function $u_0(t)$ is referred to as **Heaviside(t)**, and the delta function $\delta(t)$ is referred to as **Dirac(t)**. Their use is illustrated with the examples below.

```
syms k a t;                  % Define symbolic variables
u=k*sym('Heaviside(t–a)')    % Create unit step function at t = a
```

```
u =
k*Heaviside(t-a)
```

```
d=diff(u)                    % Compute the derivative of the unit step function
```

```
d =
k*Dirac(t-a)
```

---

* *An introduction to MATLAB® is given in Appendix A.*

Chapter 2

The Laplace Transformation

This chapter begins with an introduction to the Laplace transformation, definitions, and properties of the Laplace transformation. The initial value and final value theorems are also discussed and proved. It continues with the derivation of the Laplace transform of common functions of time, and concludes with the derivation of the Laplace transforms of common waveforms.

2.1 Definition of the Laplace Transformation

The *two–sided* or *bilateral* Laplace Transform pair is defined as

$$\boxed{\mathscr{L}\{f(t)\} = F(s) = \int_{-\infty}^{\infty} f(t)e^{-st}dt} \tag{2.1}$$

$$\boxed{\mathscr{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j}\int_{\sigma - j\omega}^{\sigma + j\omega} F(s)e^{st}ds} \tag{2.2}$$

where $\mathscr{L}\{f(t)\}$ denotes the Laplace transform of the time function $f(t)$, $\mathscr{L}^{-1}\{F(s)\}$ denotes the Inverse Laplace transform, and s is a complex variable whose real part is σ, and imaginary part ω, that is, $s = \sigma + j\omega$.

In most problems, we are concerned with values of time t greater than some reference time, say $t = t_0 = 0$, and since the initial conditions are generally known, the two–sided Laplace transform pair of (2.1) and (2.2) simplifies to the *unilateral* or *one–sided Laplace transform* defined as

$$\boxed{\mathscr{L}\{f(t)\} = F(s) = \int_{t_0}^{\infty} f(t)e^{-st}dt = \int_{0}^{\infty} f(t)e^{-st}dt} \tag{2.3}$$

$$\boxed{\mathscr{L}^{-1}\{F(s)\} = f(t) = \frac{1}{2\pi j}\int_{\sigma - j\omega}^{\sigma + j\omega} F(s)e^{st}ds} \tag{2.4}$$

The Laplace Transform of (2.3) has meaning only if the integral converges (reaches a limit), that is, if

2.2.3 Frequency Shifting Property

The *frequency shifting property* states that if we multiply a time domain function $f(t)$ by an exponential function e^{-at} where a is an arbitrary positive constant, this multiplication will produce a shift of the *s* variable in the complex frequency domain by a units. Thus,

$$\boxed{e^{-at}f(t) \Leftrightarrow F(s+a)} \tag{2.14}$$

Proof:

$$\mathcal{L}\{e^{-at}f(t)\} = \int_0^\infty e^{-at}f(t)e^{-st}dt = \int_0^\infty f(t)e^{-(s+a)t}dt = F(s+a)$$

Note 2:

A change of scale is represented by multiplication of the time variable t by a positive scaling factor a. Thus, the function $f(t)$ after scaling the time axis, becomes $f(at)$.

2.2.4 Scaling Property

Let a be an arbitrary positive constant; then, the *scaling property* states that

$$\boxed{f(at) \Leftrightarrow \frac{1}{a}F\left(\frac{s}{a}\right)} \tag{2.15}$$

Proof:

$$\mathcal{L}\{f(at)\} = \int_0^\infty f(at)e^{-st}dt$$

and letting $t = \tau/a$, we obtain

$$\mathcal{L}\{f(at)\} = \int_0^\infty f(\tau)e^{-s(\tau/a)}d\left(\frac{\tau}{a}\right) = \frac{1}{a}\int_0^\infty f(\tau)e^{-(s/a)\tau}d(\tau) = \frac{1}{a}F\left(\frac{s}{a}\right)$$

Note 3:

Generally, the initial value of $f(t)$ is taken at $t = 0^-$ to include any discontinuity that may be present at $t = 0$. If it is known that no such discontinuity exists at $t = 0^-$, we simply interpret $f(0^-)$ as $f(0)$.

2.2.5 Differentiation in Time Domain Property

The *differentiation in time domain* property states that differentiation in the time domain corresponds to multiplication by s in the complex frequency domain, minus the initial value of $f(t)$ at $t = 0^-$. Thus,

$$\boxed{f(t+nT) \Leftrightarrow \frac{\int_0^T f(t)e^{-st}dt}{1-e^{-sT}}} \tag{2.28}$$

Proof:

The Laplace transform of a periodic function can be expressed as

$$\mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st}dt = \int_0^T f(t)e^{-st}dt + \int_T^{2T} f(t)e^{-st}dt + \int_{2T}^{3T} f(t)e^{-st}dt + \dots$$

In the first integral of the right side, we let $t = \tau$, in the second $t = \tau + T$, in the third $t = \tau + 2T$, and so on. The areas under each period of $f(t)$ are equal, and thus the upper and lower limits of integration are the same for each integral. Then,

$$\mathcal{L}\{f(t)\} = \int_0^T f(\tau)e^{-s\tau}d\tau + \int_0^T f(\tau+T)e^{-s(\tau+T)}d\tau + \int_0^T f(\tau+2T)e^{-s(\tau+2T)}d\tau + \dots \tag{2.29}$$

Since the function is periodic, i.e., $f(\tau) = f(\tau+T) = f(\tau+2T) = \dots = f(\tau+nT)$, we can write (2.29) as

$$\mathcal{L}\{f(\tau)\} = (1 + e^{-sT} + e^{-2sT} + \dots)\int_0^T f(\tau)e^{-s\tau}d\tau \tag{2.30}$$

By application of the binomial theorem, that is,

$$1 + a + a^2 + a^3 + \dots = \frac{1}{1-a} \tag{2.31}$$

we find that expression (2.30) reduces to

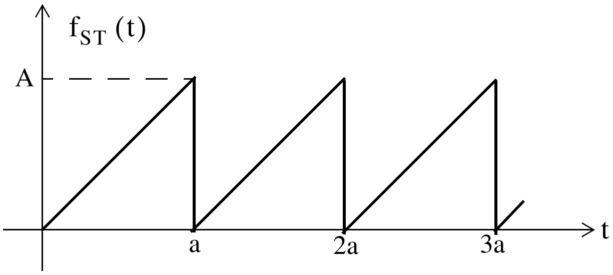
$$\mathcal{L}\{f(\tau)\} = \frac{\int_0^T f(\tau)e^{-s\tau}d\tau}{1-e^{-sT}}$$

2.2.10 Initial Value Theorem

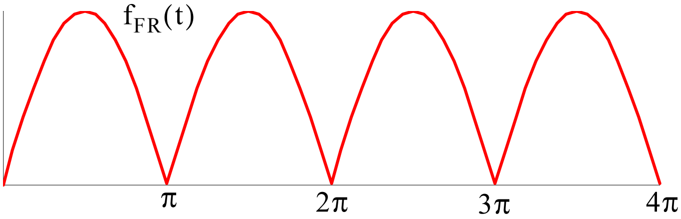
The *initial value theorem* states that the initial value $f(0^-)$ of the time function $f(t)$ can be found from its Laplace transform multiplied by s and letting $s \to \infty$. That is,

$$\boxed{\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s) = f(0^-)} \tag{2.32}$$

8. Derive the Laplace transform for the sawtooth waveform $f_{ST}(t)$ below.



9. Derive the Laplace transform for the full–rectified waveform $f_{FR}(t)$ below.



Write a simple MATLAB script that will produce the waveform above.

$$F_1(s) = \frac{3s+2}{s^2+3s+2} \tag{3.8}$$

Solution:

Using (3.6), we obtain

$$F_1(s) = \frac{3s+2}{s^2+3s+2} = \frac{3s+2}{(s+1)(s+2)} = \frac{r_1}{(s+1)} + \frac{r_2}{(s+2)} \tag{3.9}$$

The residues are

$$r_1 = \lim_{s \to -1}(s+1)F(s) = \left.\frac{3s+2}{(s+2)}\right|_{s=-1} = -1 \tag{3.10}$$

and

$$r_2 = \lim_{s \to -2}(s+2)F(s) = \left.\frac{3s+2}{(s+1)}\right|_{s=-2} = 4 \tag{3.11}$$

Therefore, we express (3.9) as

$$F_1(s) = \frac{3s+2}{s^2+3s+2} = \frac{-1}{(s+1)} + \frac{4}{(s+2)} \tag{3.12}$$

and from Table 2.2, Chapter 2, Page 2–22, we find that

$$e^{-at}u_0(t) \Leftrightarrow \frac{1}{s+a} \tag{3.13}$$

Therefore,

$$F_1(s) = \frac{-1}{(s+1)} + \frac{4}{(s+2)} \Leftrightarrow (-e^{-t} + 4e^{-2t})u_0(t) = f_1(t) \tag{3.14}$$

The residues and poles of a rational function of polynomials such as (3.8), can be found easily using the MATLAB **residue(a,b)** function. For this example, we use the script

```
Ns = [3, 2]; Ds = [1, 3, 2]; [r, p, k] = residue(Ns, Ds)
```

and MATLAB returns the values

```
r =
      4
     -1
p =
     -2
     -1
k =
      []
```

$$F_7(s) = \frac{-2s+4}{(s^2+1)(s-1)^2} \tag{3.56}$$

Solution:

Using Steps 1 through 3 above, we obtain

$$F_7(s) = \frac{-2s+4}{(s^2+1)(s-1)^2} = \frac{r_1 s + A}{(s^2+1)} + \frac{r_{21}}{(s-1)^2} + \frac{r_{22}}{(s-1)} \tag{3.57}$$

With Step 4,

$$-2s+4 = (r_1 s + A)(s-1)^2 + r_{21}(s^2+1) + r_{22}(s-1)(s^2+1) \tag{3.58}$$

and with Step 5,

$$\begin{aligned} -2s+4 &= (r_1 + r_{22})s^3 + (-2r_1 + A - r_{22} + r_{21})s^2 \\ &\quad + (r_1 - 2A + r_{22})\, s + (A - r_{22} + r_{21}) \end{aligned} \tag{3.59}$$

Relation (3.59) will be an identity is s if each power of s is the same on both sides of this relation. Therefore, we equate like powers of s and we obtain

$$\begin{aligned} 0 &= r_1 + r_{22} \\ 0 &= -2r_1 + A - r_{22} + r_{21} \\ -2 &= r_1 - 2A + r_{22} \\ 4 &= A - r_{22} + r_{21} \end{aligned} \tag{3.60}$$

Subtracting the second equation of (3.60) from the fourth, we obtain

$$4 = 2r_1$$

or

$$r_1 = 2 \tag{3.61}$$

By substitution of (3.61) into the first equation of (3.60), we obtain

$$0 = 2 + r_{22}$$

or

$$r_{22} = -2 \tag{3.62}$$

Next, substitution of (3.61) and (3.62) into the third equation of (3.60) yields

$$-2 = 2 - 2A - 2$$

or

$$A = 1 \tag{3.63}$$

Finally by substitution of (3.61), (3.62), and (3.63) into the fourth equation of (3.60), we obtain

$$\frac{4/3}{s+4}+\frac{2/3}{s+1/4} \Leftrightarrow \frac{2}{3}(2e^{-4t}+e^{-t/4})$$

d.

$$\frac{s^3+8s^2+24s+32}{s^2+6s+8}=\frac{(s+4)(s^2+4s+8)}{(s+2)(s+4)}=\frac{(s^2+4s+8)}{(s+2)} \text{ and by long division}$$

$$\frac{s^2+4s+8}{s+2}=s+2+\frac{4}{s+2} \Leftrightarrow \delta'(t)+2\delta(t)+4e^{-2t}$$

e.

$$e^{-2s}\frac{3}{(2s+3)^3} \qquad e^{-2s}F(s) \Leftrightarrow f(t-2)u_0(t-2)$$

$$F(s)=\frac{3}{(2s+3)^3}=\frac{3/2^3}{(2s+3)^3/2^3}=\frac{3/8}{[(2s+3)/2]^3}=\frac{3/8}{(s+3/2)^3} \Leftrightarrow \frac{3}{8}\left(\frac{1}{2!}t^2e^{-(3/2)t}\right)=\frac{3}{16}t^2e^{-(3/2)t}$$

$$e^{-2s}F(s)=e^{-2s}\frac{3}{(2s+3)^3} \Leftrightarrow \frac{3}{16}(t-2)^2e^{-(3/2)(t-2)}u_0(t-2)$$

4. The initial value theorem states that $\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s)$. Then,

$$f(0) = \lim_{s\to\infty} s\frac{2s+3}{s^2+4.25s+1} = \lim_{s\to\infty}\frac{2s^2+3s}{s^2+4.25s+1}$$

$$= \lim_{s\to\infty}\frac{2s^2/s^2+3s/s^2}{s^2/s^2+4.25s/s^2+1/s^2} = \lim_{s\to\infty}\frac{2+3/s}{1+4.25/s+1/s^2}=2$$

The value $f(0) = 2$ is the same as in the time domain expression that we found in Exercise 3(c).

5. We are given that $F(s) = \frac{A(s-1)}{s(s+1)}$ and $\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s) = 10$. Then,

$$\lim_{s\to 0} s\frac{A(s-1)}{s(s+1)} = A\lim_{s\to 0}\frac{(s-1)}{(s+1)} = -A = 10$$

Therefore,

$$F(s)=\frac{-10(s-1)}{s(s+1)}=\frac{r_1}{s}+\frac{r_2}{s+1}=\frac{10}{s}-\frac{20}{s+1} \Leftrightarrow (10-20e^{-t})u_0(t)$$

that is,

$$f(t) = (10-20e^{-t})u_0(t)$$

and we observe that

$$\lim_{t\to\infty} f(t) = 10$$

Chapter 4

Circuit Analysis with Laplace Transforms

This chapter presents applications of the Laplace transform. Several examples are presented to illustrate how the Laplace transformation is applied to circuit analysis. Complex impedance, complex admittance, and transfer functions are also defined.

4.1 Circuit Transformation from Time to Complex Frequency

In this section we will show the voltage–current relationships for the three elementary circuit networks, i.e., resistive, inductive, and capacitive in the time and complex frequency domains. They are shown in Subsections 4.1.1 through 4.1.3 below.

4.1.1 Resistive Network Transformation

The time and complex frequency domains for purely resistive networks are shown in Figure 4.1.

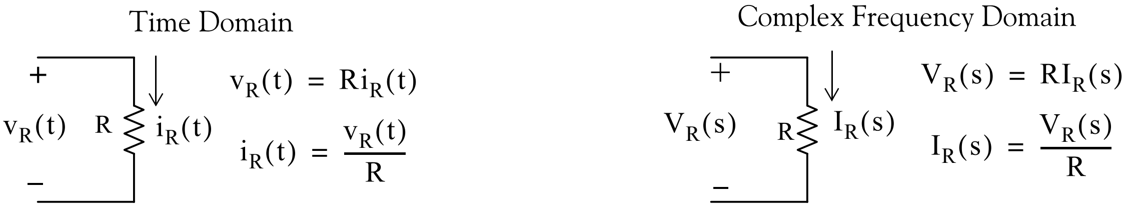


Figure 4.1. Resistive network in time domain and complex frequency domain

4.1.2 Inductive Network Transformation

The time and complex frequency domains for purely inductive networks are shown in Figure 4.2.

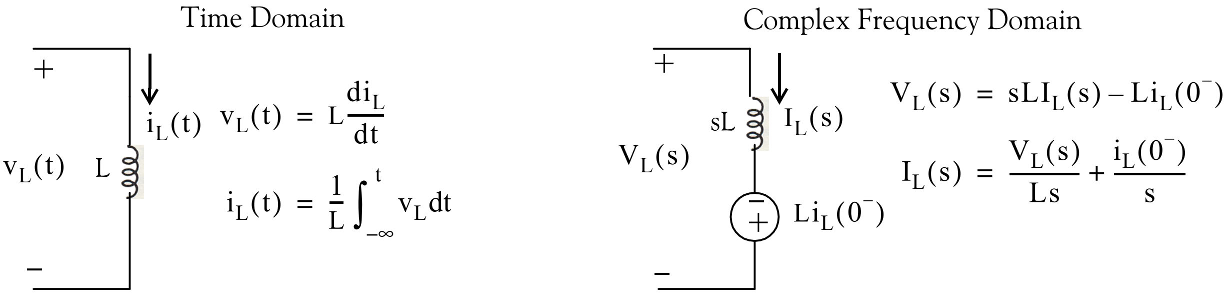

Time Domain
+
$v_L(t)$
L
$i_L(t)$
−
$v_L(t) = L\frac{di_L}{dt}$
$i_L(t) = \frac{1}{L}\int_{-\infty}^{t} v_L dt$
Complex Frequency Domain
+
$V_L(s)$
sL
$I_L(s)$
$Li_L(0^-)$
−
$V_L(s) = sLI_L(s) - Li_L(0^-)$
$I_L(s) = \frac{V_L(s)}{Ls} + \frac{i_L(0^-)}{s}$

Figure 4.2. Inductive network in time domain and complex frequency domain

4.1.3 Capacitive Network Transformation

The time and complex frequency domains for purely capacitive networks are shown in Figure 4.3.

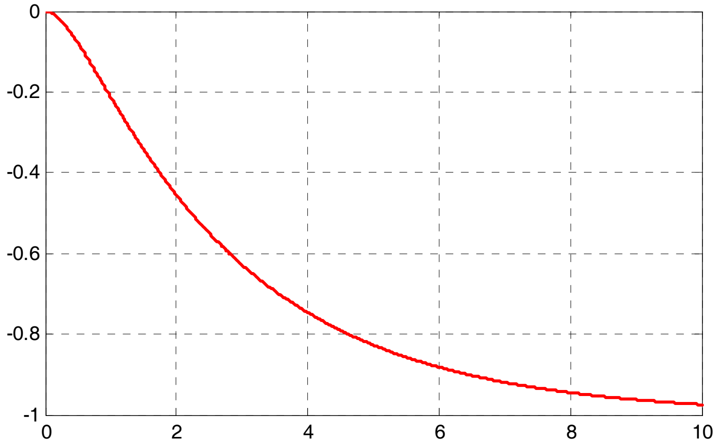


Figure 4.23. Plot of $v_{out}(t)$ for Example 4.8.

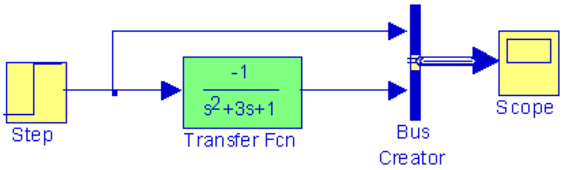


Figure 4.24. Simulink model for Example 4.8

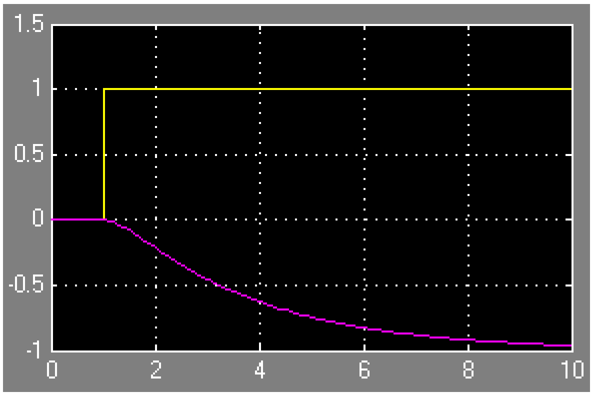
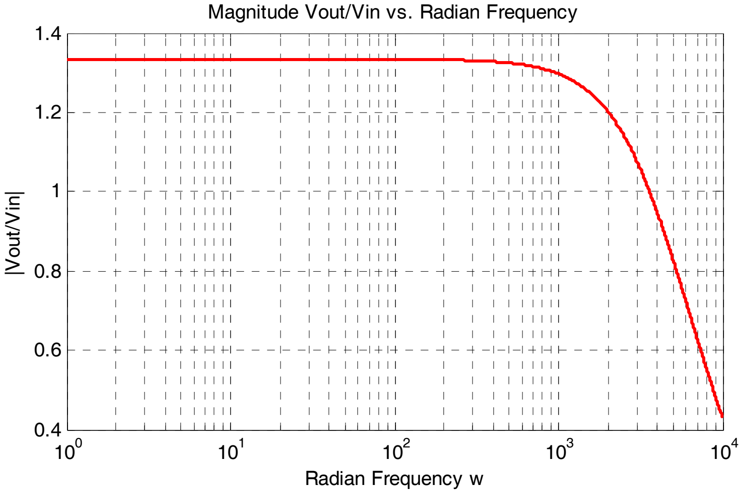


Figure 4.25. Waveform for the Simulink model of Figure 4.24



The plot above indicates that this circuit is a second–order low–pass filter.

Chapter 5

State Variables and State Equations

This chapter is an introduction to state variables and state equations as they apply in circuit analysis. The state transition matrix is defined, and the state–space to transfer function equivalence is presented. Several examples are presented to illustrate their application.

5.1 Expressing Differential Equations in State Equation Form

As we know, when we apply Kirchoff's Current Law (KCL) or Kirchoff's Voltage Law (KVL) in networks that contain energy–storing devices, we obtain integro–differential equations. Also, when a network contains just one such device (capacitor or inductor), it is said to be a *first–order circuit*. If it contains two such devices, it is said to be *second–order circuit,* and so on. Thus, a first order linear, time–invariant circuit can be described by a differential equation of the form

$$a_1\frac{dy}{dt} + a_0\, y(t) = x(t) \tag{5.1}$$

A second order circuit can be described by a second–order differential equation of the same form as (5.1) where the highest order is a second derivative.

An *nth–order* differential equation can be resolved to n first–order simultaneous differential equations with a set of auxiliary variables called *state variables*. The resulting first–order differential equations are called *state–space equations*, or simply *state equations*. These equations can be obtained either from the nth–order differential equation, or directly from the network, provided that the state variables are chosen appropriately. The state variable method offers the advantage that it can also be used with non–linear and time–varying devices. However, our discussion will be limited to linear, time–invariant circuits.

State equations can also be solved with numerical methods such as Taylor series and Runge–Kutta methods, but these will not be discussed in this text*. The state variable method is best illustrated with several examples presented in this chapter.

Example 5.1

A series RLC circuit with excitation

$$v_S(t) = e^{j\omega t} \tag{5.2}$$

---

* *These are discussed in "Numerical Analysis Using MATLAB and Excel", Third Edition, ISBN 978-1-934404-03-4.*

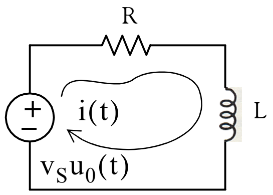


Figure 5.4. Circuit for Example 5.4

Solution:

This circuit contains only one energy–storing device, the inductor; therefore, we need only one state variable. We choose the state variable to denote the current through the inductor as shown in Figure 5.5.

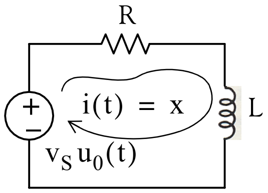


Figure 5.5. Circuit for Example 5.4 with assigned state variable x

By KVL,

$$v_R + v_L = v_S u_0(t)$$

or

$$Ri + L\frac{di}{dt} = v_S u_0(t)$$

or

$$Rx + L\dot{x} = v_S u_0(t)$$

Therefore, the state equations are

$$\begin{aligned} \dot{x} &= -\frac{R}{L}x + \frac{1}{L}v_S u_0(t) \\ y &= x \end{aligned} \tag{5.21}$$

5.2 Solution of Single State Equations

If a circuit contains only one energy–storing device, the state equations are written as

$$\begin{aligned} \dot{x} &= \alpha x + \beta u \\ y &= k_1 x + k_2 u \end{aligned} \tag{5.22}$$

where α, β, k_1, and k_2 are scalar constants, and the initial condition, if non–zero, is denoted as

$$x_0 = x(t_0) \tag{5.23}$$

$$\sqrt{2^2+1^2+3^2} = \sqrt{14}$$

$$\sqrt{1^2+1^2+2^2} = \sqrt{6}$$

$$\sqrt{(-1)^2+1^2+1^2} = \sqrt{3}$$

The unit eigenvectors are

$$\text{Unit } X_{\lambda=1} = \begin{bmatrix} \frac{2}{\sqrt{14}} \\ \frac{1}{\sqrt{14}} \\ \frac{3}{\sqrt{14}} \end{bmatrix} \qquad \text{Unit } X_{\lambda=2} = \begin{bmatrix} \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \end{bmatrix} \qquad \text{Unit } X_{\lambda=3} = \begin{bmatrix} \frac{-1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} \tag{5.83}$$

We observe that for the first unit eigenvector the sum of the squares is unity, that is,

$$\left(\frac{2}{\sqrt{14}}\right)^2 + \left(\frac{1}{\sqrt{14}}\right)^2 + \left(\frac{3}{\sqrt{14}}\right)^2 = \frac{4}{14} + \frac{1}{14} + \frac{9}{14} = 1 \tag{5.84}$$

and the same is true for the other two unit eigenvectors in (5.83).

5.6 Circuit Analysis with State Variables

In this section we will present two examples to illustrate how the state variable method is used in circuit analysis.

Example 5.10

For the circuit of Figure 5.7, the initial conditions are $i_L(0^-) = 0$, and $v_C(0^-) = 0.5\ \text{V}$. Use the state variable method to compute $i_L(t)$ and $v_C(t)$.

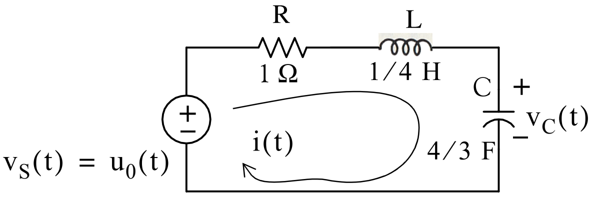


Figure 5.7. Circuit for Example 5.10

A: [−1 −1; 3/4 0]
B: [1 0]’
C: [0 1]
D: [0]
Initial conditions: [0 0]

For the **Transfer Fcn** block Function Block Parameters dialog box we have entered:

Numerator coefficient: [1]
Denominator coefficient: [1 1 1]

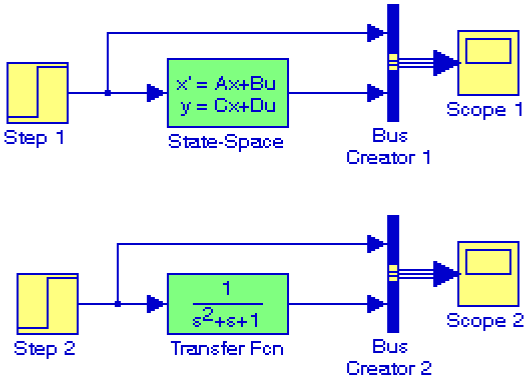


Figure 5.14. Models to show the equivalence between relations (5.126) and (5.127)

After the simulation command is executed, both Scope 1 and Scope 2 blocks display the input and output waveforms shown in Figure 5.15.

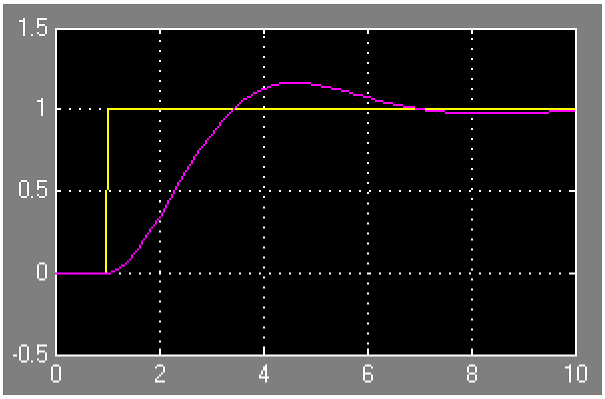


Figure 5.15. Waveforms displayed by Scope 1 and Scope 2 blocks for the models in Figure 5.14

$$e^{At} = \begin{bmatrix} -0.5e^{-t}+1.5e^{-3t} & -2e^{-t}+2e^{-3t} \\ \frac{3}{8}e^{-t}-\frac{3}{8}e^{-3t} & 1.5e^{-t}-0.5e^{-3t} \end{bmatrix}$$

The impulse response is obtained from (6.5), Page 6–1, that is,

$$h(t)= x(t) = e^{At}bu_0(t)$$

then,

$$h(t)= x(t) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.5e^{-t}+1.5e^{-3t} & -2e^{-t}+2e^{-3t} \\ \frac{3}{8}e^{-t}-\frac{3}{8}e^{-3t} & 1.5e^{-t}-0.5e^{-3t} \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix} u_0(t) = \begin{bmatrix} -2e^{-t}+6e^{-3t} \\ \frac{3}{2}e^{-t}-\frac{3}{2}e^{-3t} \end{bmatrix} u_0(t) \qquad (6.10)$$

In Example 5.10, Chapter 5, Page 5–22, we defined

$$x_1 = i_L$$

and

$$x_2 = v_C$$

Then,

$$h(t) = x_2 = v_C(t) = 1.5e^{-t} - 1.5e^{-3t}$$

or

$$h(t) = v_C(t) = 1.5(e^{-t} - e^{-3t}) \qquad (6.11)$$

Of course, this answer is not the same as that of Example 5.10, because the inputs and initial conditions were defined differently.

6.2 Even and Odd Functions of Time

A function $f(t)$ is an *even function* of time if the following relation holds.

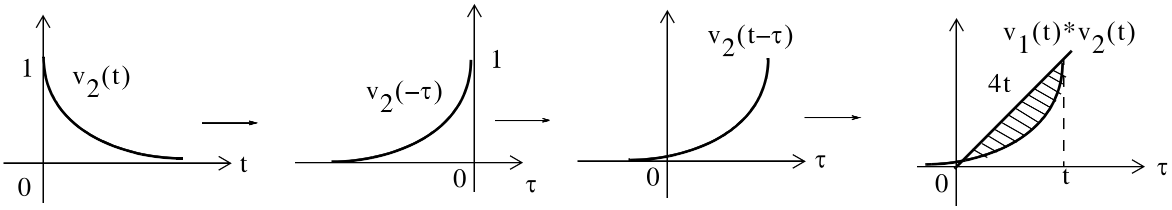
$$\boxed{f(-t) = f(t)} \qquad (6.12)$$

that is, if in an even function we replace t with $-t$, the function $f(t)$ does not change. Thus, polynomials with even exponents only, and with or without constants, are even functions. For instance, the cosine function is an even function because it can be written as the power series

$$\cos t = 1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \frac{t^6}{6!} + \dots$$

Other examples of even functions are shown in Figure 6.4.

4.



$$v_1(t)*v_2(t) = \int_0^t v_1(\tau)v_2(t-\tau)d\tau = \int_0^t 4\tau e^{-2(t-\tau)}d\tau = 4e^{-2t}\int_0^t \tau e^{2\tau}d\tau$$

From tables of integrals,

$$\int xe^{ax}dx = \frac{e^{ax}}{a^2}(ax-1)$$

and thus

$$v_1(t)*v_2(t) = (4e^{-2t})\frac{e^{2\tau}(2\tau-1)}{4}\Bigg|_0^t = e^{-2t}[e^{2t}(2t-1)-e^0(-1)]$$

$$= e^0(2t-1+e^{-2t}) = 2t+e^{-2t}-1$$

Check:

$$v_1(t)*v_2(t) \Leftrightarrow V_1(s)\cdot V_2(s)\,,\; V_1(s) = 4/s^2\,,\; V_2(s) = 1/(s+2)$$

$$V_1(s)\cdot V_2(s) = \frac{4}{s^2(s+2)} = \frac{4}{s^3+2s^2}$$

```
syms s t; ilaplace(4/(s^3+2*s^2))
```

```
ans =
2*t-1+exp(-2*t)
```

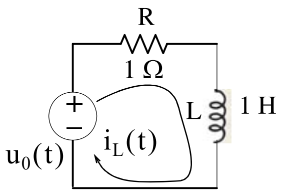
5.

To use the convolution integral, we must first find the impulse response. It was found in Exercise 1 as

$$h(t) = i(t) = (1/L)e^{-(R/L)t}u_0(t)$$

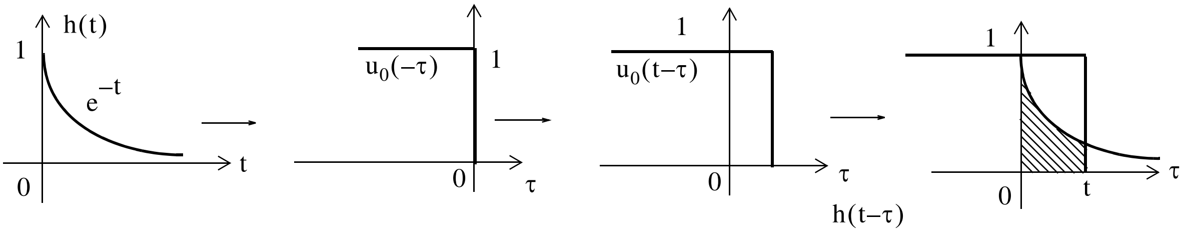
and with the given values,

$$h(t) = e^{-t}u_0(t)$$

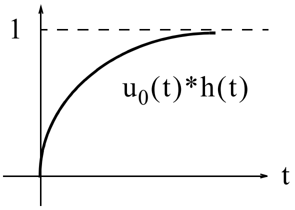


When the input is the unit step $u_0(t)$,

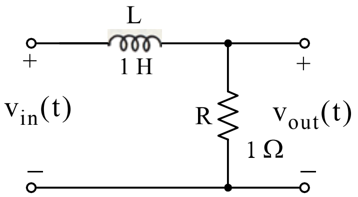
$$i(t)\big|_{v_{in} = u_0(t)} = \int_{-\infty}^{\infty} u_0(t-\tau)h(\tau)d\tau$$



$$i(t)\big|_{v_{in} = u_0(t)} = \int_0^t (1)\cdot e^{-\tau}d\tau = -e^{-\tau}\Big|_0^t = e^{-\tau}\Big|_t^0 = (1-e^{-t})u_0(t)$$



6.



We will first compute the impulse response, that is, the output when the input is the delta function, i.e., $v_{in}(t) = \delta(t)$. Then, by KVL

$$L\frac{di_L}{dt} + Ri_L = \delta(t)$$

and with $i_L = x$

$$1\cdot\dot{x} + 1\cdot x = \delta(t)$$

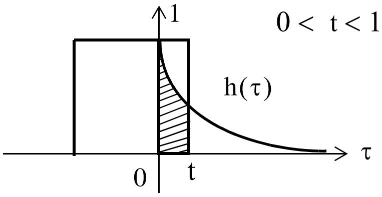
or

$$\dot{x} = -x + \delta(t)$$

By comparison with $\dot{x} = Ax + bu$, we observe that $A = -1$ and $b = 1$.
From (6.5)

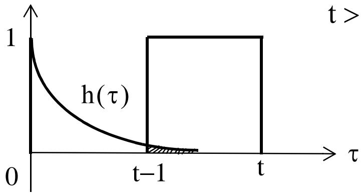
$$h(t) = e^{At}bu_0(t) = e^{-t} \cdot 1 = e^{-t}$$

Now, we compute $v_{out}(t)$ when $v_{in}(t) = u_0(t) - u_0(t-1)$ by convolving the impulse response $h(t)$ with this input $v_{in}(t)$, that is, $v_{out}(t) = v_{in}(t)*h(t)$. The remaining steps are as in Example 6.5 and are shown below.



$0 < t < 1$

$$\int_0^t (1)(e^{-\tau})d\tau = -e^{-\tau}\Big|_0^t = e^{-\tau}\Big|_t^0 = 1 - e^{-t}$$

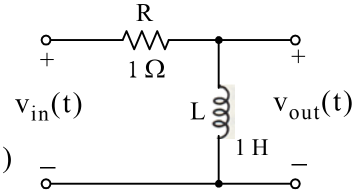


$t > 1$

$$\int_{t-1}^t (1)(e^{-\tau})d\tau = -e^{-\tau}\Big|_{t-1}^t = e^{-\tau}\Big|_t^{t-1} = e^{-t}(e-1)$$

7.

$$v_{in}(t) = u_0(t) - u_0(t-1)$$



$$v_{out}(t) = v_L = v_{in} - v_R$$

From Exercise 6,

$$v_R = \begin{cases} 1 - e^{-t} & 0 < t < 1 \\ e^{-t}(e-1) & t > 1 \end{cases}$$

Then, for this circuit,

$$v_{out} = v_L = \begin{cases} (1 - (1 - e^{-t}) = e^{-t}) & 0 < t < 1 \\ 0 - e^{-t}(e-1) = (1-e)e^{-t} = -1.732e^{-t} & t > 1 \end{cases}$$

The plot for the time interval $0 < t < 5$ is shown below.

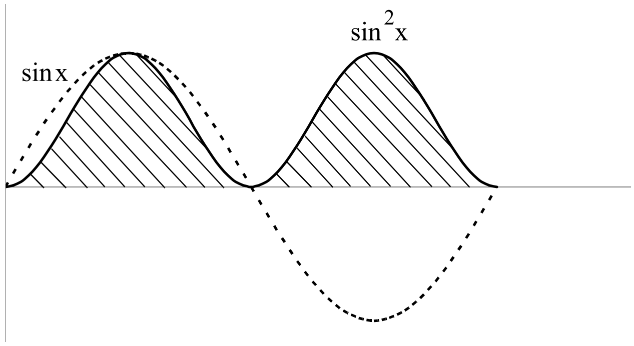


Figure 7.5. *Graphical proof of* $\int_0^{2\pi} (\sin mt)^2 dt = \pi$

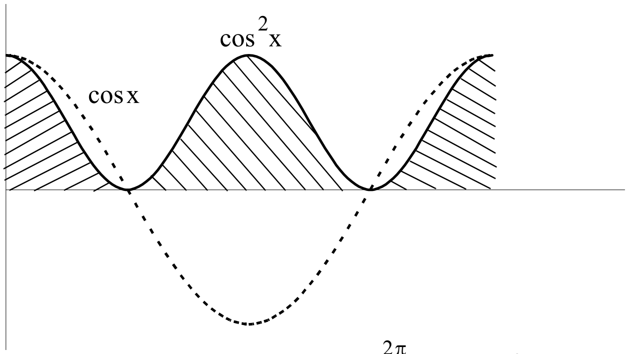


Figure 7.6. *Graphical proof of* $\int_0^{2\pi} (\cos mt)^2 dt = \pi$

To evaluate any coefficient in (7.10), say b_2, we multiply both sides of (7.10) by $\sin 2t$. Then,

$$f(t)\sin 2t = \frac{1}{2}a_0 \sin 2t + a_1 \cos t \sin 2t + a_2 \cos 2t \sin 2t + a_3 \cos 3t \sin 2t + a_4 \cos 4t \sin 2t + \dots$$

$$b_1 \sin t \sin 2t + b_2 (\sin 2t)^2 + b_3 \sin 3t \sin 2t + b_4 \sin 4t \sin 2t + \dots$$

Next, we multiply both sides of the above expression by dt, and we integrate over the period 0 to 2π. Then,

$$\begin{aligned}\int_0^{2\pi} f(t)\sin 2t dt &= \frac{1}{2}a_0 \int_0^{2\pi} \sin 2t dt + a_1 \int_0^{2\pi} \cos t \sin 2t dt + a_2 \int_0^{2\pi} \cos 2t \sin 2t dt \\ &+ a_3 \int_0^{2\pi} \cos 3t \sin 2t dt + \dots \\ &+ b_1 \int_0^{2\pi} \sin t \sin 2t dt + b_2 \int_0^{2\pi} (\sin 2t)^2 dt + b_3 \int_0^{2\pi} \sin 3t \sin 2t dt + \dots\end{aligned} \qquad (7.11)$$

that is, the shape of the negative half–cycle of the waveform is the same as that of the positive half-cycle, but inverted.

We will test the most common waveforms for symmetry in Subsections 7.3.1 through 7.3.5 below.

7.3.1 Symmetry in Square Waveform

For the waveform of Figure 7.7, the average value over one period T is zero, and therefore, $a_0 = 0$. It is also an odd function and has half–wave symmetry since $-f(-t) = f(t)$ and $-f(t + T/2) = f(t)$.

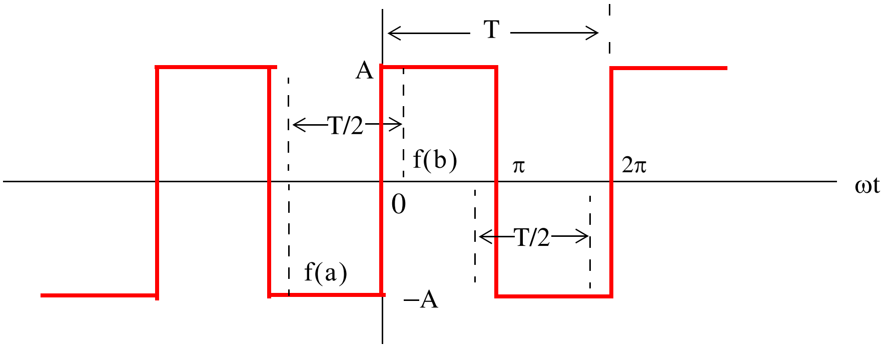


Figure 7.7. Square waveform test for symmetry

An easy method to test for half–wave symmetry is to choose any half–period $T/2$ length on the time axis as shown in Figure 7.7, and observe the values of $f(t)$ at the left and right points on the time axis, such as $f(a)$ and $f(b)$. If there is half–wave symmetry, these will always be equal but will have opposite signs as we slide the half-period $T/2$ length to the left or to the right on the time axis at non–zero values of $f(t)$.

7.3.2 Symmetry in Square Waveform with Ordinate Axis Shifted

If in the square waveform of Figure 7.7 we shift the ordinate axis $\pi/2$ radians to the right, as shown in Figure 7.8, we will observe that the square waveform now becomes an even function, and has half–wave symmetry since $f(-t) = f(t)$ and $-f(t + T/2) = f(t)$. Also, $a_0 = 0$.

Obviously, if the ordinate axis is shifted by any other value other than an odd multiple of $\pi/2$, the waveform will have neither odd nor even symmetry.

The waveform of Figure 7.30 is a rudimentary presentation of the capacitor voltage for the circuit of Figure 7.27. However, it will improve if we add a sufficient number of harmonics in (7.87).

We can obtain a more accurate waveform for the capacitor voltage of Figure 7.27 with the Simulink model of Figure 7.31.

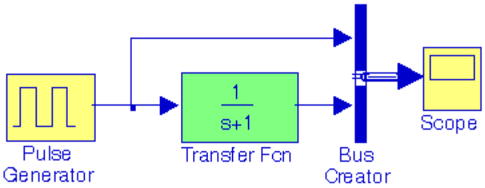


Figure 7.31. Simulink model for the circuit of Figure 7.27

The input and output waveforms are shown in Figure 7.32.

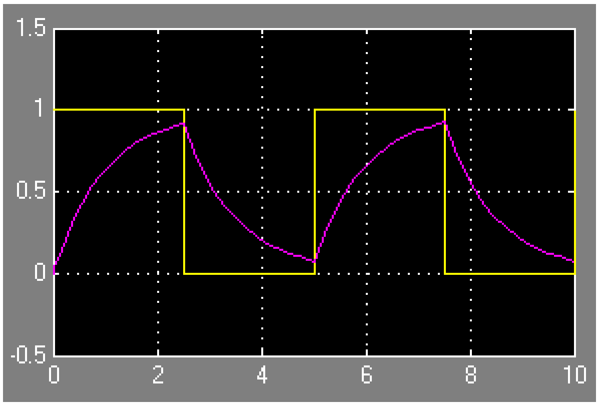


Figure 7.32. Input and output waveforms for the model of Figure 7.31

7.8 The Exponential Form of the Fourier Series

The Fourier series are often expressed in exponential form. The advantage of the exponential form is that we only need to perform one integration rather than two, one for the a_n, and another for the b_n coefficients in the trigonometric form of the series. Moreover, in most cases the integration is simpler.

The exponential form is derived from the trigonometric form by substitution of

$$\cos\omega t = \frac{e^{j\omega t} + e^{-j\omega t}}{2} \tag{7.88}$$

$$\sin\omega t = \frac{e^{j\omega t} - e^{-j\omega t}}{j2} \tag{7.89}$$

8.2 Special Forms of the Fourier Transform

The time function $f(t)$ is, in general, complex, and thus we can express it as the sum of its real and imaginary parts, that is, as

$$f(t) = f_{Re}(t) + j f_{Im}(t) \tag{8.6}$$

The subscripts Re and Im will be used often to denote the real and imaginary parts respectively. These notations have the same meaning as $Re\{f(t)\}$ and $Im\{f(t)\}$.

By substitution of (8.6) into the Fourier integral of (8.1), we obtain

$$F(\omega) = \int_{-\infty}^{\infty} f_{Re}(t)e^{-j\omega t}dt + j\int_{-\infty}^{\infty} f_{Im}(t)e^{-j\omega t}dt \tag{8.7}$$

and by Euler's identity

$$F(\omega) = \int_{-\infty}^{\infty}[f_{Re}(t)\cos\omega t + f_{Im}(t)\sin\omega t]dt - j\int_{-\infty}^{\infty}[f_{Re}(t)\sin\omega t - f_{Im}(t)\cos\omega t]dt \tag{8.8}$$

From (8.8), we see that the real and imaginary parts of $F(\omega)$ are

$$F_{Re}(\omega) = \int_{-\infty}^{\infty}[f_{Re}(t)\cos\omega t + f_{Im}(t)\sin\omega t]dt \tag{8.9}$$

and

$$F_{Im}(\omega) = -\int_{-\infty}^{\infty}[f_{Re}(t)\sin\omega t - f_{Im}(t)\cos\omega t]dt \tag{8.10}$$

We can derive similar forms for the Inverse Fourier transform as follows:

Substitution of (8.2) into (8.3) yields

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[F_{Re}(\omega) + jF_{Im}(\omega)]e^{j\omega t}d\omega \tag{8.11}$$

and by Euler's identity,

$$\begin{aligned} f(t) = {} & \frac{1}{2\pi}\int_{-\infty}^{\infty}[F_{Re}(\omega)\cos\omega t - F_{Im}(\omega)\sin\omega t]d\omega \\ & + j\frac{1}{2\pi}\int_{-\infty}^{\infty}[F_{Re}(\omega)\sin\omega t + F_{Im}(\omega)\cos\omega t]d\omega \end{aligned} \tag{8.12}$$

Therefore, the real and imaginary parts of $f(t)$ in terms of the Inverse Fourier transform are

$$\boxed{f_{Re}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[F_{Re}(\omega)\cos\omega t - F_{Im}(\omega)\sin\omega t]d\omega} \tag{8.13}$$

and

TABLE 8.3 Time Domain and Frequency Domain Correspondence (Refer also to Tables 8.4 – 8.7)

f(t)	**F(ω)**				
	Real	**Imaginary**	**Complex**	**Even**	**Odd**
Real			✓		
Real and Even	✓			✓	
Real and Odd					
Imaginary					
Imaginary and Even					
Imaginary and Odd					

$$\boxed{F_{Re}(\omega) = \int_{-\infty}^{\infty} f_{Re}(t)\cos\omega t\,dt = 0 \qquad f_{Re}(t) = odd} \tag{8.20}$$

and

$$\boxed{F_{Im}(\omega) = -2\int_{0}^{\infty} f_{Re}(t)\sin\omega t\,dt \qquad f_{Re}(t) = odd} \tag{8.21}$$

Therefore, if $f_{Re}(t) = odd$, $F(\omega)$ is imaginary.

To determine whether $F(\omega)$ is even or odd when $f_{Re}(t) = odd$, we perform a test for evenness or oddness with respect to ω. Thus, substitution of $-\omega$ for ω in (8.21), yields

$$F_{Im}(-\omega) = -2\int_{0}^{\infty} f_{Re}(t)\sin(-\omega)t\,dt = 2\int_{0}^{\infty} f_{Re}(t)\sin\omega t\,dt = -F_{Im}(\omega) \tag{8.22}$$

Conclusion: If $f(t)$ is real and odd, $F(\omega)$ is imaginary and odd. We indicate this result in Table 8.4.

TABLE 8.4 Time Domain and Frequency Domain Correspondence (Refer also to Tables 8.5 – 8.7)

f(t)	**F(ω)**				
	Real	**Imaginary**	**Complex**	**Even**	**Odd**
Real			✓		
Real and Even	✓			✓	
Real and Odd		✓			✓
Imaginary					
Imaginary and Even					
Imaginary and Odd					

Example 8.5

$$te^{-t^2} \Leftrightarrow \frac{1}{2}j\sqrt{\pi}\omega e^{-\frac{1}{4}\omega^2} \tag{8.98}$$

syms t v w x; ft=t*exp(–t^2); Fw=fourier (ft)

```
Fw =
  -1/2*i*pi^(1/2)*w*exp(-1/4*w^2)
```

pretty(Fw)

```
                    1/2              2
        - 1/2 i pi      w exp(- 1/4 w )
```

Example 8.6

$$-e^{-t}u_0(t) + 3\delta(t) \Leftrightarrow -\frac{1}{j\omega + 1} + 3 \tag{8.99}$$

syms t v w x; fourier(sym('–exp(–t)*Heaviside(t)+3*Dirac(t)'))

```
ans =
  -1/(1+i*w)+3
```

Example 8.7

$$u_0(t) \Leftrightarrow \pi\delta(\omega) + \frac{1}{j\omega} \tag{8.100}$$

syms t v w x; u0=sym('Heaviside(t)'); Fw=fourier(u0)

```
Fw =
pi*Dirac(w)-i/w
```

We have summarized the most common Fourier transform pairs in Table 8.9.

8.8 The System Function and Applications to Circuit Analysis

We recall from Chapter 6 that, by definition, the convolution integral is

$$h(t)*u(t) = \int_{-\infty}^{\infty} u(t-\tau)h(\tau)d\tau \tag{8.101}$$

We let

$$g(t) = f(t)*h(t) \tag{8.102}$$

We could also have used numerical integration with the integral

$$\int_0^{\pi} \frac{\sin^2 x}{x^2} dx$$

thereby avoiding the integration by parts procedure. Shown below are the function `fourierxfm2` which is saved as an `.m` file and the program execution using this function.

```
function y2=fourierxfm2(x)
x=x+(x==0)*eps;
y2=(sin(x)./x).^2;
```

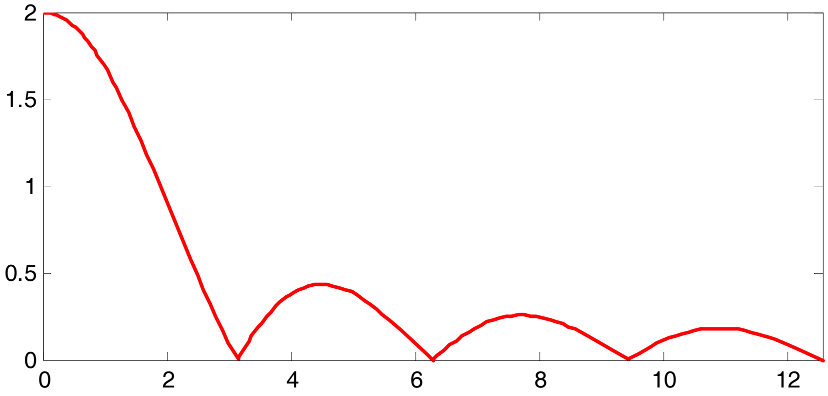
and after this file is saved, we execute the statement below observing that the limits of integration are from 0 to π.

```
value2=quad('fourierxfm2',0,pi)
```

```
value2 =
  1.4182
```

11.

```
fplot('abs(2.*exp(–j.*w)*(sin(w)/w))',[0 4*pi 0 2])
```



and for $m = 2$, reduces to

$$\boxed{\mathcal{Z}\{f[n+2]\} = z^2F(z) - f[0]z^2 - f[1]z} \tag{9.16}$$

9.2.5 Multiplication by a^n in the Discrete–Time Domain

$$\boxed{a^n f[n] \Leftrightarrow F\left(\frac{z}{a}\right)} \tag{9.17}$$

Proof:

$$\mathcal{Z}\{a^n f[n]\} = \sum_{k=0}^{\infty} a^n f[n] z^{-n} = \sum_{k=0}^{\infty} \frac{1}{a^{-n}} f[n] z^{-n} = \sum_{k=0}^{\infty} f[n]\left(\frac{z}{a}\right)^{-n} = F\left(\frac{z}{a}\right)$$

9.2.6 Multiplication by e^{-naT} in the Discrete–Time Domain

$$\boxed{e^{-naT} f[n] \Leftrightarrow F(e^{aT}z)} \tag{9.18}$$

Proof:

$$\mathcal{Z}\{e^{-naT} f[n]\} = \sum_{k=0}^{\infty} e^{-naT} f[n] z^{-n} = \sum_{k=0}^{\infty} f[n](e^{aT}z)^{-n} = F(e^{aT}z)$$

9.2.7 Multiplication by n and n^2 in the Discrete–Time Domain

$$\boxed{\begin{aligned} nf[n] &\Leftrightarrow -z\frac{d}{dz}F(z) \\ n^2 f[n] &\Leftrightarrow z\frac{d}{dz}F(z) + z^2\frac{d^2}{dz^2}F(z) \end{aligned}} \tag{9.19}$$

Proof:

By definition,

$$F(z) = \sum_{n=0}^{\infty} f[n] z^{-n}$$

and taking the first derivative of both sides with respect to z, we obtain

$$\frac{d}{dz}F(z) = \sum_{n=0}^{\infty} (-n) f[n] z^{-n-1} = -z^{-1} \sum_{n=0}^{\infty} nf[n] z^{-n}$$

$$Y(z) = \mathcal{Z}\left\{\sum_{m=0}^{\infty} x[m]h[n-m]\right\} = \sum_{n=0}^{\infty}\left[\sum_{m=0}^{\infty} x[m]h[n-m]\right]z^{-n}$$

and interchanging the order of the summation, we obtain

$$Y(z) = \sum_{m=0}^{\infty}\left[\sum_{n=0}^{\infty} x[m]h[n-m]z^{-n}\right] = \sum_{m=0}^{\infty} x[m]\sum_{n=0}^{\infty} h[n-m]z^{-n}$$

Next, we let $k = n-m$, then, $n = k+m$, and thus,

$$Y(z) = \sum_{m=0}^{\infty} x[m]\sum_{n=0}^{\infty} h[k]z^{-(k+m)} = \sum_{m=0}^{\infty} x[m]z^{-m}\sum_{n=0}^{\infty} h[k]z^{-k}$$

or

$$Y(z) = X(z)\cdot H(z) \tag{9.27}$$

9.2.10 Convolution in the Discrete–Frequency Domain

If $f_1[n]$ and $f_2[n]$ are two sequences with $\mathcal{Z}$ transforms $F_1(z)$ and $F_2(z)$ respectively, then,

$$f_1[n]\cdot f_2[n] \Leftrightarrow \frac{1}{j2\pi}\oint xF_1(v)F_2\left(\frac{z}{v}\right)v^{-1}dv \tag{9.28}$$

where v is a dummy variable, and $\oint$ is a closed contour inside the overlap convergence regions for $X_1(v)$ and $X_2(z/v)$. The proof requires contour integration; it will not be provided here.

9.2.11 Initial Value Theorem

$$\boxed{f[0] = \lim_{z\to\infty} X(z)} \tag{9.29}$$

Proof:

For all $n \ge 1$, as $z \to \infty$

$$z^{-n} = \frac{1}{z^n} \to 0$$

and under these conditions $f[n]z^{-n} \to 0$ also. Taking the limit as $z \to \infty$ in

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n}$$

```
num=[0  1  0  0];                    % The coefficients of the numerator
den=[1  –9/4   13/8   –3/8];         % The coefficients of the denominator
fprintf(' \n');
[num,den]=residue(num,den);     % Verify the residues in (9.79)
fprintf('r1 = %4.2f \t', num(1)); fprintf('p1 = %4.2f \t', den(1));...
fprintf('r2 = %4.2f \t', num(2)); fprintf('p2 = %4.2f \t', den(2));...
fprintf('r3 = %4.2f \t', num(3)); fprintf('p3 = %4.2f \t', den(3))
```

```
 r1 = 8.00 p1 = 1.00   r2 = -9.00 p2 = 0.75   r3 = 2.00 p3 = 0.50
```

```
syms n z
fn=2*(0.5)^n–9*(0.75)^n+8;          % This is the answer in (9.80)
Fz=ztrans(fn,n,z); simple(Fz)       % Verify answer by first taking Z transform of f[n]
```

```
ans =
 8*z^3/(2*z-1)/(4*z-3)/(z-1)
```

```
iztrans(Fz)                         % Now, verify that Inverse of F(z) gives back f[n]
```

```
ans =
 2*(1/2)^n-9*(3/4)^n+8
```

We can use Microsoft Excel to obtain and plot the values of $f[n]$. The spreadsheet of Figure 9.6 shows the first 25 values of n but only part of the spreadsheet is shown.

n	f[n]
0.000	1.0000
1.000	2.2500
2.000	3.438
3.000	4.453
4.000	5.277
5.000	5.927
6.000	6.429
7.000	6.814
8.000	7.107
9.000	7.328
10.000	7.495
11.000	7.621
12.000	7.715
13.000	7.786
14.000	7.84
15.000	7.88

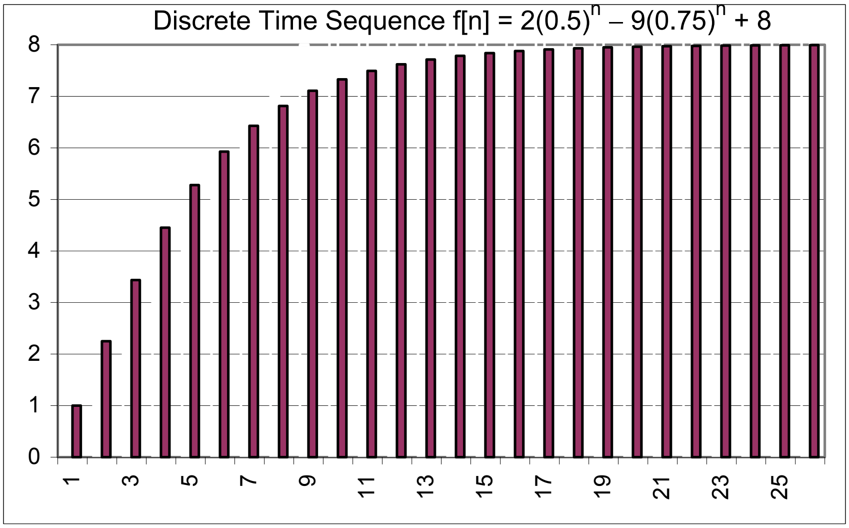


Figure 9.6. The discrete–time sequence $f[n] = 2(0.5)^n - 9(0.75)^n + 8$ for Example 9.4

or

$$f[n] = 16 + \frac{(163/64)(0.75)^n}{(-0.25)(0.75)^2} = 16 - \frac{163}{9}(0.75)^n$$

for $n \ge 2$.

We can express $f[n]$ for all $n \ge 0$ as

$$f[n] = \frac{28}{9}\delta[n] + \frac{4}{3}\delta[n-1] + 16 - \frac{163}{9}(0.75)^n \tag{9.96}$$

where the coefficients of $\delta[n]$ and $\delta[n-1]$ are the residues that were found in (9.92) and (9.93) for $n = 0$ and $n = 1$ respectively at $z = 0$. The coefficient $28/9$ is multiplied by $\delta[n]$ to emphasize that this value exists only for $n = 0$ and coefficient $4/3$ is multiplied by $\delta[n-1]$ to emphasize that this value exists only for $n = 1$.

Check with MATLAB:

```
syms z n; Fz=(z^3+2*z^2+1)/(z*(z–1)*(z–0.75)); iztrans(Fz)
```

```
ans =
   4/3*charfcn[1](n)+28/9*charfcn[0](n)+16-163/9*(3/4)^n
```

We evaluate and plot $f[n]$ for the first 20 values. This is shown on the spreadsheet of Figure 9.9.

n	f[n]
0	1.000
1	3.750
2	5.813
3	8.359
4	10.270
5	11.702
6	12.777
7	13.582
8	14.187
9	14.640
10	14.980
11	15.235
12	15.426
13	15.570

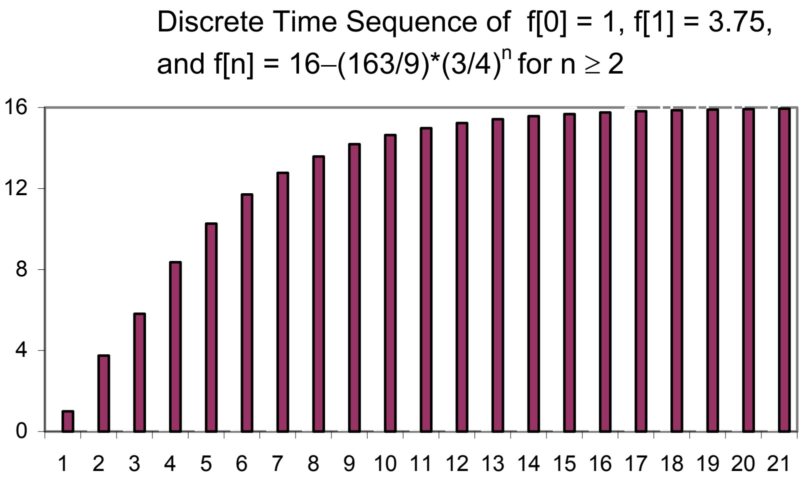


Figure 9.9. The discrete–time sequence for Example 9.7

Example 9.8

Use the inversion integral method to find the Inverse $\mathcal{Z}$ transform of

$$F(z) = \sum_{n=0}^{\infty} f[n]z^{-n} \tag{10.1}$$

Its value on the unit circle of the z-plane, is

$$F(e^{j\omega T}) = \sum_{n=0}^{\infty} f[n]e^{-nj\omega T} \tag{10.2}$$

This represents an infinite summation; it must be truncated before it can be computed. Let this truncated version be represented by

$$\boxed{X[m] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{mn}{N}}} \tag{10.3}$$

where N represents the number of points that are equally spaced in the interval 0 to 2π on the unit circle of the z-plane, and

$$\omega = \left(\frac{2\pi}{NT}\right)m$$

for $m = 0, 1, 2, \ldots, N-1$. We refer to relation (10.3) as the $N-\text{point}$ DFT of $X[m]$.

The Inverse DFT is defined as

$$\boxed{x[n] = \frac{1}{N}\sum_{m=0}^{N-1} X[m]e^{j2\pi\frac{mn}{N}}} \tag{10.4}$$

for $n = 0, 1, 2, \ldots, N-1$.

In general, the discrete frequency transform $X[m]$ is complex, and thus we can express it as

$$X[m] = \mathrm{Re}\{X[m]\} + \mathrm{Im}\{X[m]\} \tag{10.5}$$

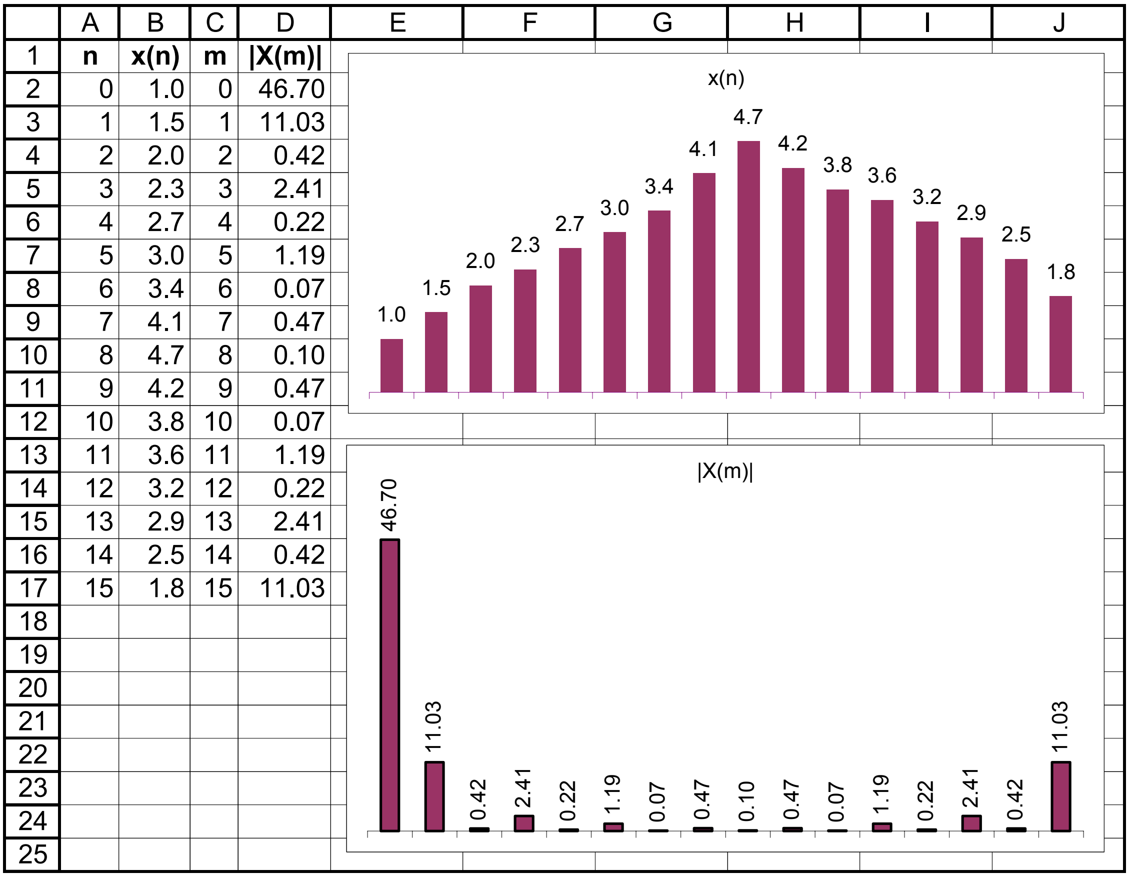
for $m = 0, 1, 2, \ldots, N-1$.

Since

$$e^{-j2\pi\frac{mn}{N}} = \cos\frac{2\pi mn}{N} - j\sin\frac{2\pi mn}{N} \tag{10.6}$$

we can express (10.3) as

$$X[m] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi\frac{mn}{N}} = \sum_{n=0}^{N-1} x[n]\cos\frac{2\pi mn}{N} - j\sum_{n=0}^{N-1} x[n]\sin\frac{2\pi mn}{N} \tag{10.7}$$



	A	B	C	D
1	n	x(n)	m	|X(m)|
2	0	1.0	0	46.70
3	1	1.5	1	11.03
4	2	2.0	2	0.42
5	3	2.3	3	2.41
6	4	2.7	4	0.22
7	5	3.0	5	1.19
8	6	3.4	6	0.07
9	7	4.1	7	0.47
10	8	4.7	8	0.10
11	9	4.2	9	0.47
12	10	3.8	10	0.07
13	11	3.6	11	1.19
14	12	3.2	12	0.22
15	13	2.9	13	2.41
16	14	2.5	14	0.42
17	15	1.8	15	11.03
18				
19				
20				
21				
22				
23				
24				
25				

Figure 10.2. Plots of $x[n]$ *and* $|X[m]|$ *values for Example 10.3*

On the plot of $|X[m]|$ in Figure 10.2, the first value $X[0] = 46.70$ represents the DC component. We observe that after the $X[8] = 0.10$ value, the magnitude of the frequency components for the range $9 \le m \le 15$, are the mirror image of the components in the range $1 \le m \le 7$. This is not a coincidence; it is a fact that *if* $x[n]$ *is an* N*–point real discrete time function, only* $N/2$ *of the frequency components of* $|X[m]|$ *are unique.*

Figure 10.3 shows typical discrete time and frequency magnitude waveforms, for a $N = 16$–point DFT.

The sequences of (10.65) and (10.66) cannot be decomposed further. They justify the statement made earlier, that each computation produces a vector where each component of this vector, for $n = 1, 2, 3, \ldots, 7$ is obtained by one multiplication and one addition. This is often referred to as a *butterfly operation.*

Substitution of (10.65) and (10.66) into (10.60), yields

$$F_{even}[m] = F_{even1}[m] + W_N^{2m} F_{even2}[m] \tag{10.67}$$

Likewise, $F_{odd}[m]$ can be decomposed into DFTs of length 2; then, $F[m]$ can be computed from

$$F(m) = F_{even}(m) + W_N^m F_{odd}(m) \quad m = 0, 1, 2, \ldots, 7 \tag{10.68}$$

for $N = 8$. Of course, this procedure can be extended for any N that is divisible by 2.

Figure 10.5 shows the *signal flow graph* of a decimation in time, in–place FFT algorithm for $N = 8$, where the input is shuffled in accordance with the above procedure. The subscript N in W has been omitted for clarity.

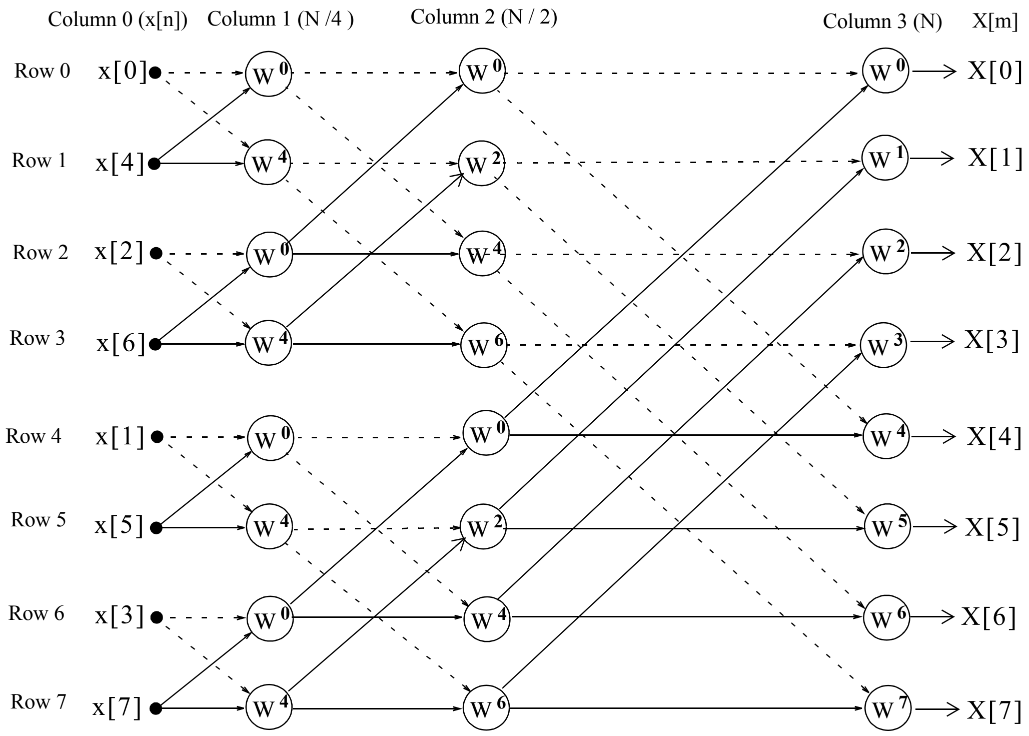
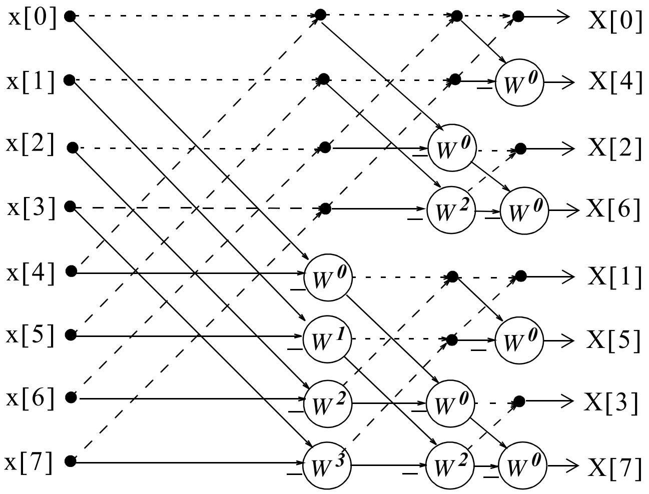
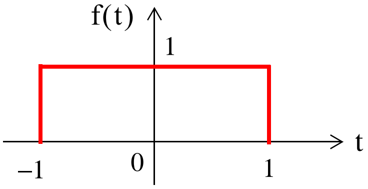


Figure 10.5. Signal flow graph of a decimation in time, in-place FFT algorithm, for $N = 8$

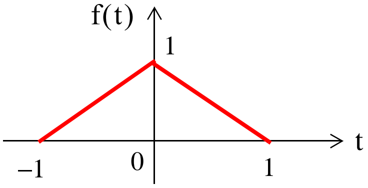
and it is used with the nodes where two solid lines terminate on them. The number inside the circles denote the power of W_N, and the minus (–) sign below serves as a reminder that the bracketed term of the second equation involves a subtraction. Using this graph and the above equations, compute the frequency component $X[3]$. Verify that this is the same as in Example 10.5.



6. Plot the Fourier transform of the rectangular pulse shown below, using the MATLAB **fft** function. Then, use the **ifft** function to verify that the inverse transformation produces the rectangular pulse.



7. Plot the Fourier transform of the triangular pulse shown below using the MATLAB **fft** function. Then, use the **ifft** function to verify that the inverse transformation produces the rectangular pulse.



11.2.3 RLC Band–Pass Filter

The RLC network shown in Figure 11.8 is a basic analog band–pass filter.* We will derive expressions for its magnitude and phase.

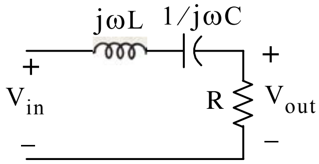


Figure 11.8. Basic band–pass RLC network

Application of the voltage division expression yields

$$V_{out} = \frac{R}{j\omega L + 1/j\omega C + R} \cdot V_{in}$$

and thus

$$G(j\omega) = \frac{V_{out}}{V_{in}} = \frac{R}{j\omega L + 1/j\omega C + R} = \frac{j\omega RC}{-\omega^2 LC + 1 + j\omega RC} = \frac{-j\omega(R/L)}{\omega^2 - j\omega(R/L) - 1/LC} \qquad (11.7)$$

There is no need to express relation (11.7) in magnitude and phase form. We will make use of the **abs(x)** and **angle(x)** MATLAB functions for the magnitude and phase plots.

We will use the MATLAB script below to plot $|G(j\omega)|$ versus radian frequency ω. This is shown in Figure 11.9 where, for convenience, we let $R = L = C = 1$, and thus (11.7) simplifies to:

$$G(j\omega) = \frac{-j\omega}{\omega^2 - j\omega - 1} \qquad (11.8)$$

```
w=0:0.02:100;  magGjw=abs(–j.*w./(w.^2–j.*w–1)); semilogx(w,magGjw);...
xlabel('Frequency in rad/sec – log scale'); ylabel('Magnitude of Vout/Vin');...
title('Magnitude Characteristics of basic RLC band–pass filter'); grid
```

The plot for the magnitude of (11.8) is shown in Figure 11.9.

To plot the phase of (11.8), we use the MATLAB script below.

```
w=0:0.02:100;  phaseGjw=angle(–j.*w./(w.^2–j.*w–1)).*180./pi; semilogx(w,phaseGjw);...
xlabel('Frequency in rad/sec – log scale'); ylabel('Phase of Vout/Vin – degrees');...
title('Phase Characteristics of basic RLC band–pass filter'); grid
```

The plot for the phase of (11.8) is shown in Figure 11.10.

---

* *This is the same network as (a) in Exercise 7, Chapter 4, Page 4–22.*

```
xlabel('Frequency in rad/sec – log scale'); ylabel('Phase of Vout/Vin – degrees');...
title('Phase Characteristics of basic RLC band–elimination filter'); grid
```

The plot for the phase of (11.10) is shown in Figure 11.13.

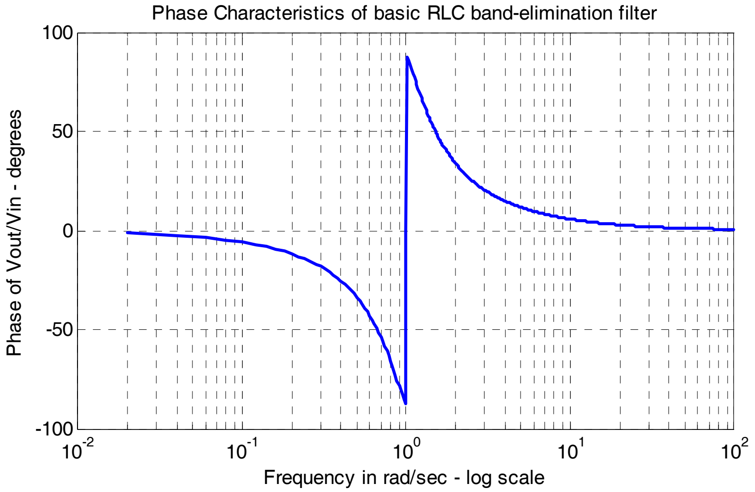


Figure 11.13. Phase characteristics of the basic RLC band–elimination filter with $R = L = C = 1$

Other passive filter networks are presented as end–of–chapter exercises in Chapter 4. Examples of active low–pass filters are presented in Figure 4.20, Chapter 4, Page 4–15, and at the end of this chapter. In the next section we introduce filter prototypes and use these as a basis for filter design.

11.3 Low–Pass Analog Filter Prototypes

In this section, we will use the analog low–pass filter as a basis. We will see later that, using transformations, we can derive high–pass and the other types of filters from a basic low–pass filter. We will discuss the *Butterworth, Chebyshev Type I, Chebyshev Type II also known as Inverted Chebyshev*, and *Cauer also known as elliptic* filters.

The first step in the design of an analog low–pass filter is to derive a suitable *magnitude–squared function* $A^2(\omega)$, and from it derive a $G(s)$ function such that

$$A^2(\omega) = G(s) \cdot G(-s)|_{s = j\omega} \qquad (11.11)$$

Since $(j\omega)^* = (-j\omega)$, the square of the magnitude of a complex number can be expressed as that number and its complex conjugate. Thus, if the magnitude is A, then

$$A^2(\omega) = |G(j\omega)| = G(j\omega)G^*(j\omega) = G(j\omega) \cdot G(-j\omega) \qquad (11.12)$$

Now, $G(j\omega)$ can be considered as $G(s)$ evaluated at $s = j\omega$, and thus (11.11) is justified. Also, since A is understood to represent the magnitude, it needs not be enclosed in vertical lines.

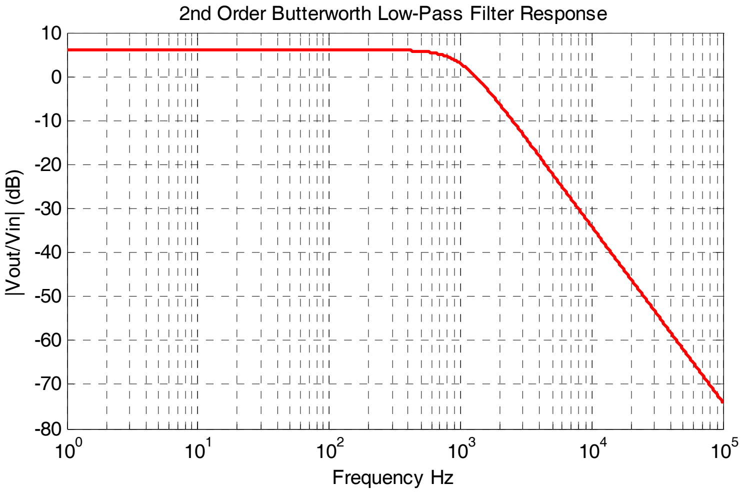


Figure 11.20. Plot for the VCVS low–pass filter of Example 11.6

We have used the MATLAB **buttap** function earlier to aid us in the design of Butterworth filters with the cutoff frequency normalized to 1 rad/s. We can also use the **bode** function to display both the (asymptotic) magnitude and phase plots. The following script will produce the Bode magnitude and phase plots for a two–pole Butterworth low-pass filter.

```
[z,p,k]= buttap(2);                    % Specify a two–pole filter;...
[b,a]=zp2tf(z,p,k);                    % Display in polynomial rational form;...
w=0:0.01:4; [mag,phase]=bode(b,a,w);...
b,a                                    % Display b and a coefficients
```

```
b =
     0     0     1

a =
    1.0000    1.4142    1.0000
```

```
num=[0  0  1]; den=[1  sqrt(2)  1];...
bode(num,den); title('Butterworth 2nd Order Low–Pass Filter'); grid
```

The Bode plots are shown in Figure 11.21. The frequency is displayed in rad/sec and the cutoff frequency normalized to 1 rad/s.

We can also display the Bode plots with the frequency specified in Hz. This can be done with the MATLAB script below.

```
h=bodeplot(tf(num,den));...
setoptions(h,'FreqUnits', 'Hz'); grid
```

The Bode plots with the frequency specified in Hz are shown in Figure 11.22.

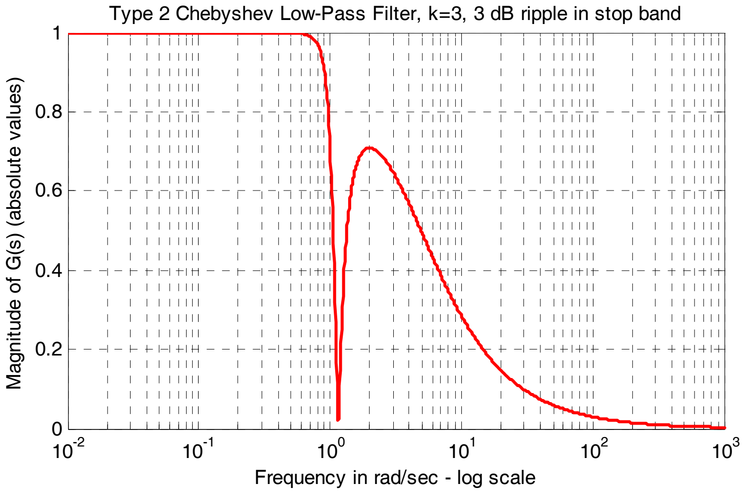


Figure 11.29. Plot for the Chebyshev Type II filter of Example 11.10

11.3.4 Elliptic Analog Low-Pass Filter Design

The *elliptic*, also known as *Cauer* filters, are characterized by the low–pass magnitude–squared function

$$A^2(\omega) = \frac{1}{1 + R_k^2(\omega/\omega_C)} \tag{11.69}$$

where $R_k(x)$ represents a rational elliptic function used with elliptic integrals. Elliptic filters have ripple in both the pass–band and the stop–band as shown in Figure 11.30.

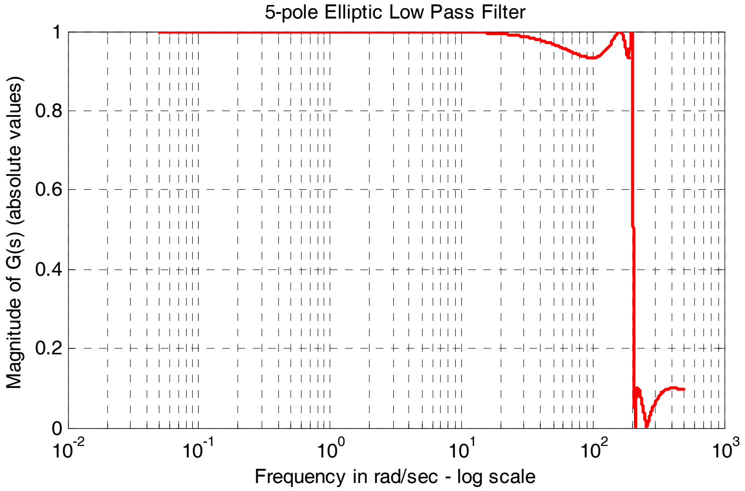


Figure 11.30. Characteristics of an elliptic low–pass filter

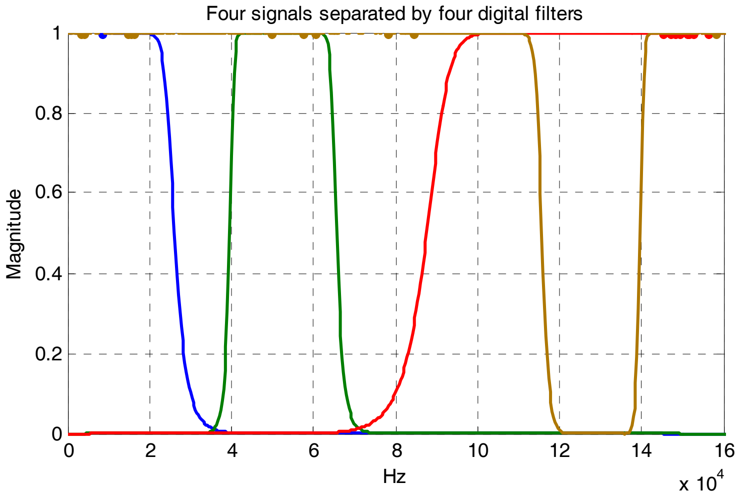


Figure 11.43. Frequency separations for the signals of Example 11.21

In the following example, we will demonstrate the MATLAB **filter** function that is being used to remove unwanted frequency components from a function. But before we use the **filter** function, we must design a filter that is capable of removing those unwanted components.

Example 11.22

In Chapter 7, Subsection 7.4.4, Page 7–20, we found that the half–wave rectifier with no symmetry can be represented by the trigonometric Fourier series

$$f(t) = \frac{A}{\pi} + \frac{A}{2}\sin t - \frac{A}{\pi}\left[\frac{\cos 2t}{3} + \frac{\cos 4t}{15} + \frac{\cos 6t}{35} + \frac{\cos 8t}{63} + \dots\right]$$

In this example, we want to filter out just the first 2 terms, in other words, to remove all cosine terms. To simplify this expression, we let $A = 3\pi$ and we truncate it by eliminating all cosine terms except the $\cos 2t$ term. Then,

$$g(t) = 3 + 1.5\sin t - \cos 2t \qquad (11.99)$$

The problem now reduces to design a low–pass digital filter, and use the **filter** command to remove the cosine term in (11.99).

Solution:

We will use a $6-\text{pole}$ digital low–pass Butterworth filter because we must have a sharp transition between the 1 and $2\ \text{rad/s}$ frequency range. Also, since the highest frequency component is $2\ \text{rad/s}$, to avoid aliasing, we must specify a sampling frequency of $\omega_S = 4\ \text{rad/s}$. Thus, the

The analog and digital inputs and outputs are shown in Figures 11.45 and 11.46 respectively.

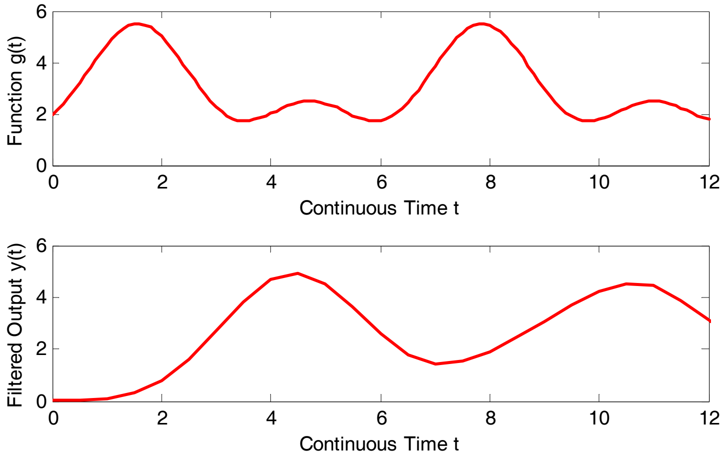


Figure 11.45. Continuous time input and output waveforms for Example 11.22

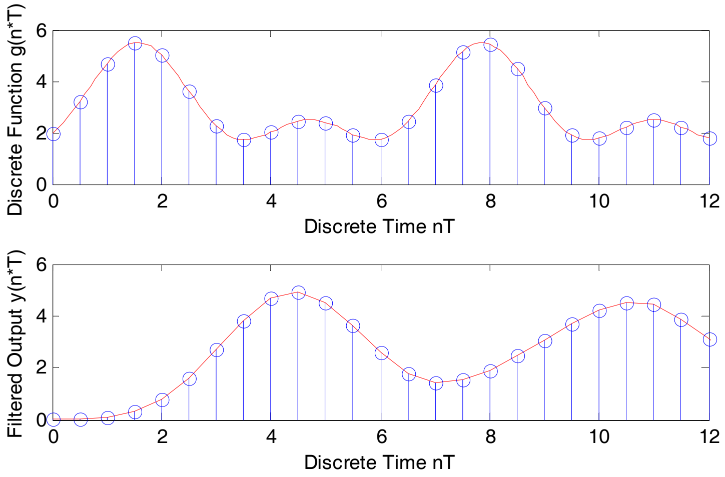


Figure 11.46. Discrete time input and output waveforms for Example 11.22

We conclude this section with one more example to illustrate the use of the MATLAB **find** function. This function displays the subscripts of an array of numbers where a relational expression is true. For example,

```
x=–2:5; % Display the integers in the range -2 <= x < =5
x =
    -2     -1      0      1      2      3      4      5
```

```
k=find(x>0.8); % Find the subscripts of the numbers for which x > 0.8
```

To implement this filter, we factor the numerator and denominator polynomials as

$$H(z) = \frac{0.5(1+0.6z^{-1})(1-0.6z^{-1})^{*}}{(1+0.9z^{-1})(1-0.8z^{-1})} \tag{11.106}$$

The model is shown in Figure 11.54, and the input and output waveforms are shown in Figure 11.55.

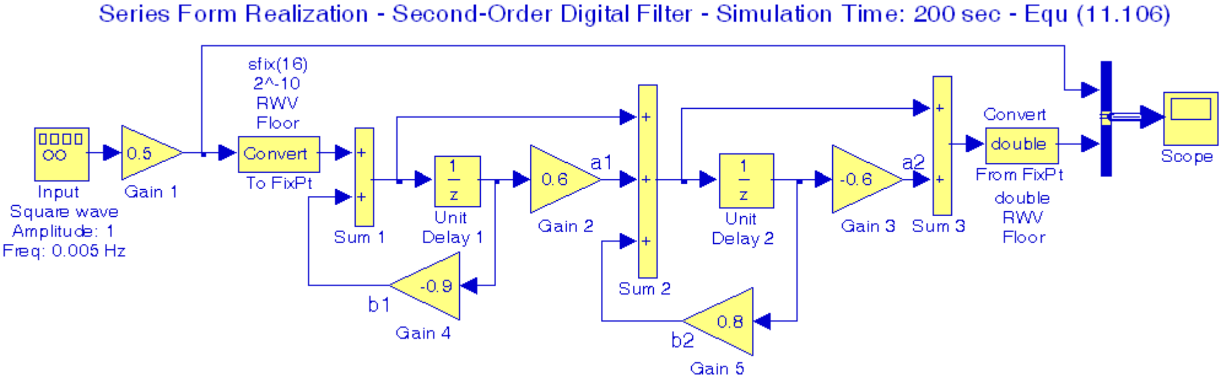


Figure 11.54. Model for Example 11.25

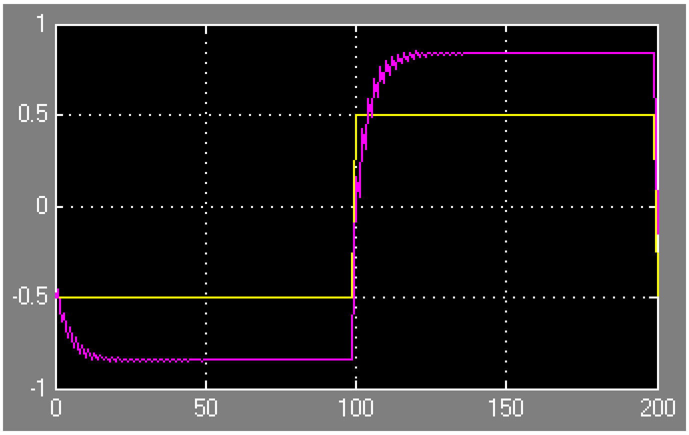


Figure 11.55. Input and output waveforms for the model of Figure 11.54

A demo model using fixed–point Simulink blocks can be displayed by typing

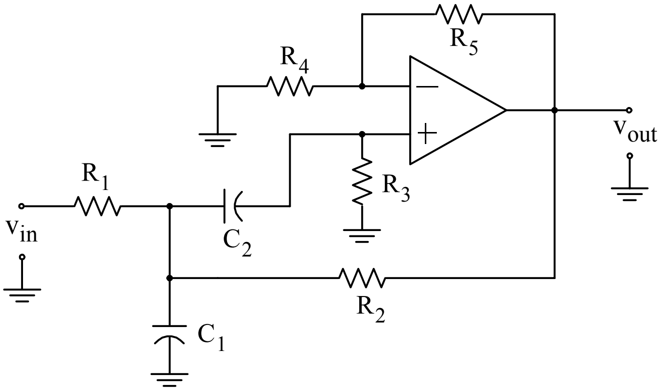
```
fxpdemo_series_cascade_form
```

---

* *The combination of the of factors in parentheses is immaterial. For instance, we can group the factors as* $\frac{(1+0.6z^{-1})}{(1+0.9z^{-1})}$ *and* $\frac{(1-0.6z^{-1})}{(1-0.8z^{-1})}$ *or as* $\frac{(1+0.6z^{-1})}{(1-0.8z^{-1})}$ *and* $\frac{(1-0.6z^{-1})}{(1+0.9z^{-1})}$

2. The op amp circuit below is a VCVS second–order band–pass filter whose transfer function is

$$G(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{K[BW]s}{s^2 + [BW]s + \omega_0^2}$$



Let ω_0 = center frequency, ω_2 = upper cutoff frequency, ω_1 = lower cutoff frequency, Bandwidth $BW = \omega_2 - \omega_1$, and Quality Factor $Q = \omega_0/BW$

We can calculate the values of C_1, C_2, R_1, R_2, R_3, and R_4 to achieve the desired centered frequency ω_0 and bandwidth BW. For this circuit,

$$R_1 = \frac{2Q}{C_1\omega_0 K}$$

$$R_2 = \frac{2Q}{C_1\omega_0\left\{-1 + \sqrt{(K-1)^2 + 8Q^2}\right\}}$$

$$R_3 = \frac{1}{C_1^2\omega_0^2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)$$

$$R_4 = R_5 = 2R_3$$

Using these relations, compute the appropriate values of the resistors to achieve center frequency $f_0 = 1\ KHz$, Gain $K = 10$, and $Q = 10$.

Choose the capacitors as $C_1 = C_2 = 0.1\ \mu F$. Plot $|G(s)|$ versus frequency.

Solution using MATLAB is highly recommended.

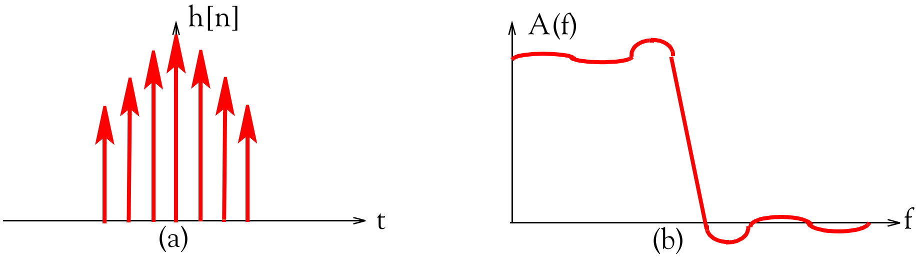


Figure E.2. Resultant amplitude response when the impulse response is abruptly truncated

Therefore, we are seeking a suitable window function that will result in an amplitude response with a lower ripple in the stop band such as the one shown in Figure E.3.

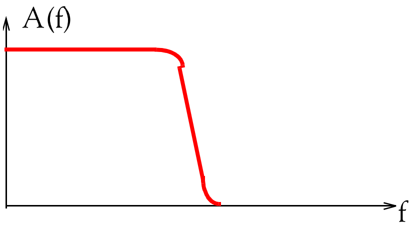


Figure E.3. Acceptable amplitude response

Although an impulse response may converge uniformly, when it is multiplied* by a rectangular window function it results in an undesirable amplitude response. However, the rectangular window function is the basis in studying window functions, so we begin with the description of the rectangular window function.

E.2.1 Rectangular Window Function

The rectangular window function is defined as

$$f(t)_{rect} = 1 \quad \text{for} \quad |t| < \frac{\tau}{2} \qquad\qquad (E.1)$$
$$= 0 \quad \text{otherwise}$$

and its time–domain waveform is as shown in Figure E.4.

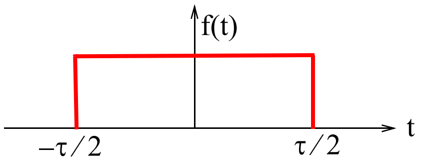


Figure E.4. Rectangular window function in time–domain

---

* *We recall that multiplication in the time domain corresponds to convolution in the frequency domain.*

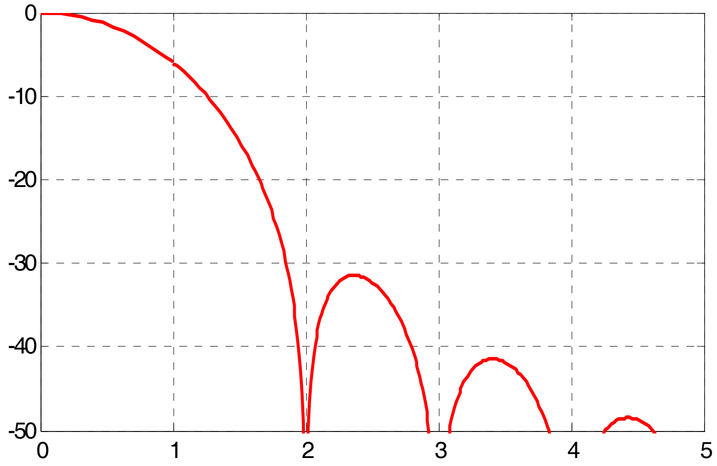


Figure E.13. Typical amplitude response for the Hanning window function

Figure E.14 shows the normalized frequency domain plot for the Hanning window **function** created with the **hann** MATLAB function, and the script below.

```
n=50; w=hann(n); [W,f]=freqz(w/sum(w),1,512,2); plot(f,20*log10(abs(W)));grid
```

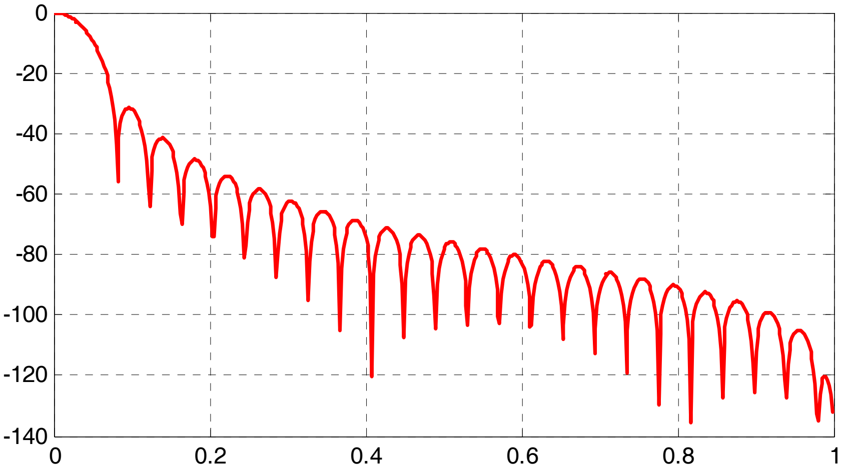


Figure E.14. Normalized frequency domain plot for the Hanning window function created with MATLAB

We can also use the MATLAB Window Visualization Tool. The script below generates the plot shown in Figure E.15.

```
L=50; wvtool(hann(L))
```

smaller side lobe levels.

MATLAB provides the Kaiser window function as

```
w=kaiser(L, beta)
```

and this returns an L–point Kaiser window* in the column vector w.

Figure E. 24 shows the Kaiser functions for $L = 50$ with $\beta = 1$, $\beta = 4$, and $\beta = 9$. These functions were plotted with the MATLAB script below.

```
plot([kaiser(50, 1), kaiser(50,4), kaiser(50,9)]);
```

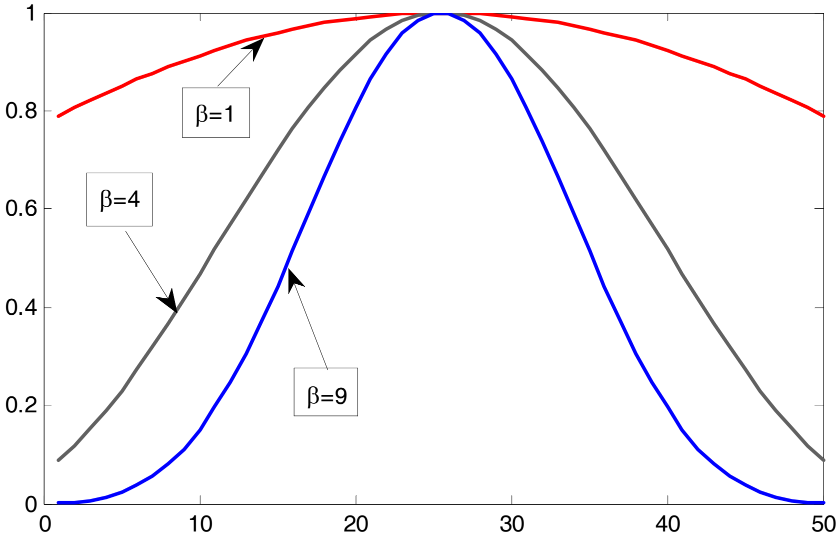


Figure E.24. Kaiser window functions with L=50 and $\beta = 1$, $\beta = 4$, *and* $\beta = 9$

Figure E.25 shows the corresponding frequency domain plots for the Kaiser window functions in Figure E.24. These plots were created with the MATLAB script below.

```
n=50;
w1 =kaiser(n, 1); w2=kaiser(n,4); w3=kaiser(n,9);
[W1,f]=freqz(w1/sum(w1),1,512,2); [W2,f]=freqz(w2/sum(w2),1,512,2);
[W3,f]=freqz(w3/sum(w3),1,512,2); plot(f,20*log10(abs([W1 W2 W3])));grid;
```

E.3 Other Window Functions

Table E.1 lists the window functions we've discussed in the previous sections, as well as others along with the MATLAB function names and are included in the MATLAB Signal Processing Toolbox.

---

* *When the expression under the radical in* (E.12) *is negative, the function can be expressed in terms of the hyperbolic sine function.*

$$H_1(z) = \sum_{m=-M}^{M} c_m z^m \tag{E.16}$$

Relation (E.16) represents an FIR transfer function but it is non–causal because it includes positive powers of z, and this implies that the filter would produce an output advanced in time with respect to the input, and this is, of course, impossible. We can be overcome this problem by introducing a delay of M samples. Accordingly, we define the transfer function

$$H(z) = z^{-M}H_1(z) = z^{-M}\sum_{m=-M}^{M} c_m z^m = \sum_{m=M}^{-M} c_m z^{m-M} \tag{E.17}$$

In (E.17) we let $m - M = -i$. Then, with the substitution $c_{m-i} = a_i$ we obtain

$$H(z) = \sum_{i=0}^{2M} c_{m-i}\, z^{-i} = \sum_{i=0}^{2M} a_i z^{-i} \tag{E.18}$$

Let w_m denote the coefficients of a particular window function and let c'_m denote the coefficients of the modified function, i.e., the given function multiplied by some window function. The modified coefficients are then computed from

$$c'_m = w_m c_m \tag{E.19}$$

Example E.1

A low–pass FIR digital filter is to be designed using the Fourier series method. The desired amplitude response is

$$A(f) = \begin{cases} 1 & \text{for } 0 \le f \le 125 \text{ Hz} \\ 0 & \text{elsewhere in the range } 0 < f < f_0 \end{cases} \tag{E.20}$$

The sampling frequency is 1 KHz and the impulse response is to be limited to 20 delays. Derive the transfer function using the Fourier series method.

Solution:

To make this low–pass filter an even function of frequency, we represent it as shown in Figure E.26 (a) in terms of the actual frequency. Let us express it in terms of the normalized frequency scale. We do this by considering the sampling frequency which is given as $f_S = 1$ KHz. Accordingly, the folding frequency[*] is $f_0 = f_S/2 = 1000/2 = 500$ Hz, and in terms of the normalized

---

[*] *We recall that the folding frequency is the highest frequency that can be processed by a given discrete–time system with sampling frequency* f_S

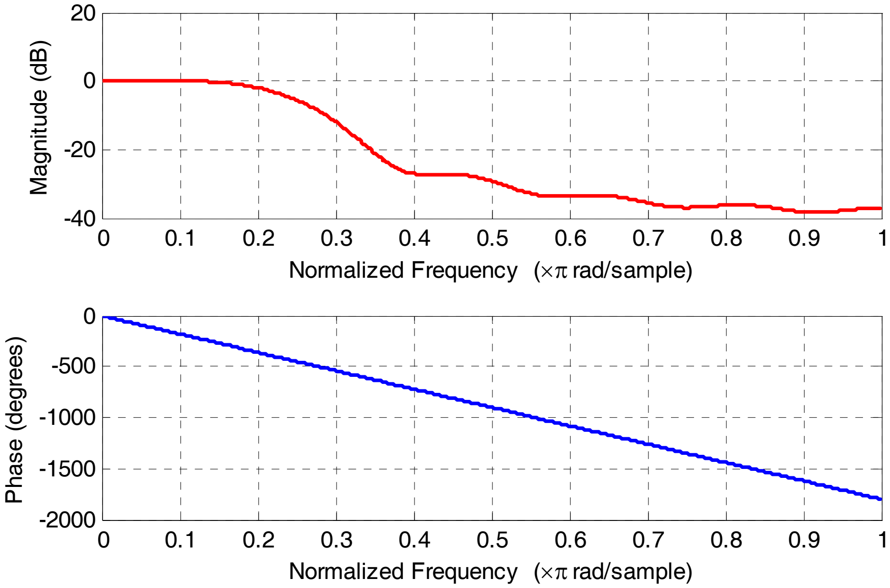


Figure E.31. Normalized frequency plots for the triang *window created with the MATLAB function* fir1

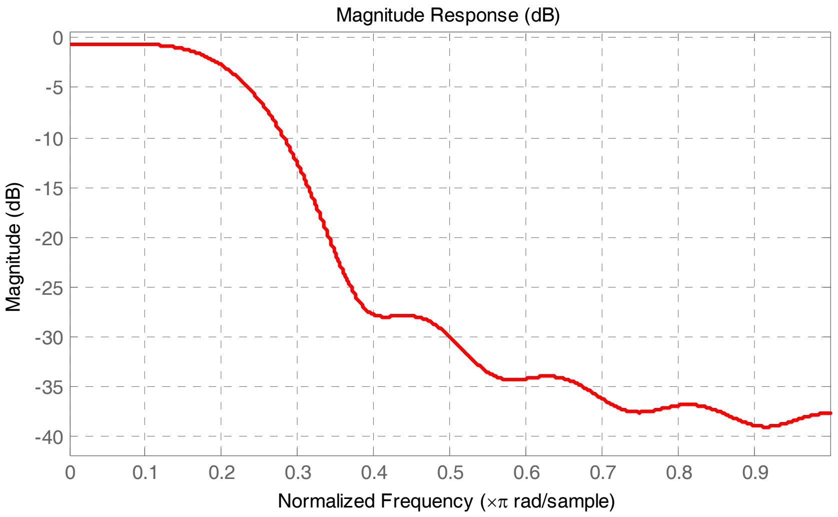


Figure E.32. Magnitude response for the triang *window created with the MATLAB function* fvtool

b. Hanning

We recall from (E.6) that the Hanning window function is defined as

$$\begin{aligned} f(t)_{\text{Hann}} &= \cos^2(\pi t/\tau) = \frac{1}{2}\left(1+\cos\frac{2\pi t}{\tau}\right) \qquad \text{for } |t| < \frac{\tau}{2} \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{E.27}$$

Index

Symbols

% (percent) symbol in MATLAB A-2

A

abs(z) in MATLAB A-23
active analog filter - see filter
adjoint of a matrix - see matrix
admittance
  capacitive 4-2
  inductive 4-2
  complex input 4-11
algebraic constrain block in Simulink B-18
aliasing 10-14
all-pass filter - see filter
all-pole approximation 11-21
all-pole low-pass filter see filter - low-pass
alternate form of the trigonometric Fourier series - see Fourier series
alternate method of partial fraction expansion - see partial fraction expansion
angle(z) MATLAB function A-23
argument 11-2
attenuation rate 11-12
autoscale icon in Simulink B-12
axis MATLAB command A-16

B

band-elimination filter - see filter
band-elimination filter design see filter design
band-limited signal 10-13
band-pass filter - see filter
band-pass filter design see filter design
band-stop filter - see filter
Bessel filter - see filter
bilinear MATLAB function 11-59
bilinear transformation - see transformation methods for mapping analog prototype filters to digital filters
bode MATLAB function 11-24
box MATLAB command A-12
buttap MATLAB function 11-17
buttefly operation 10-23
Butterworth analog low-pass filter design - see filter design

C

c2d MATLAB function 9-46
capacitive admittance - see admittance
capacitive impedance - see impedance
cascade form realization - see digital filter
Category **I** FFT algorithm - see FFT algorithm
Category **II** FFT algorithm - see FFT algorithm
Cauchy's residue theorem see residue theorem
Cauer filter - see elliptic filter
Cayley-Hamilton theorem 5-11
characteristic equation 5-19
cheb1ap MATLAB function 11-35
cheb2ap MATLAB function 11-38
Chebyshev filters - see filter
Chebyshev Type I analog low-pass filter design - see filter design
Chebyshev Type I filters - see filter
Chebyshev Type I low-pass filter magnitude-square function 11-26 - see filter
Chebyshev Type II analog low-pass filter design - see filter design
Chebyshev Type II filters - see filter
circuit analysis with Laplace transforms 4-1
circuit analysis with state variables 5-22
circuit transformation from time to complex frequency 4-1
clc MATLAB command A-2
clear MATLAB command A-2
cofactor of a matrix - see matrix
collect(s) MATLAB symbolic function 3-12
column vector in MATLAB A-19
command screen in MATLAB A-1
Command Window in MATLAB A-1
commas in MATLAB A-8
comment line in MATLAB A-2
Commonly Used Blocks in Simulink B-7
complex admittance - see admittance
complex conjugate in MATLAB A-4
complex conjugate pairs 3-5
complex impedance - see impedance
complex number C-2
complex numbers in MATLAB A-3
complex poles 3-5
Complex to Magnitude-Angle block in Simulink C-7
computation of the state transition matrix 5-11
computation of the Z Transform with contour integration - see Z transform
Configuration Parameters in Simulink B-12
congugate of a matrix - see matrix
conj(A) MATLAB function D-9
conjugate of a complex number C-3
conjugate time and frequency functions of the Fourier transform - see Fourier transform - properties of
constant function - Fourier transform of see Fourier transform - properties of
Contents Pane in Simulink B-7
contour integral 9-20
conv MATLAB function A-7
convolution in the complex frequency domain - see Laplace transform properties of
convolution in the discrete-frequency domain - see Z transform - properties of
convolution in the discrete-time domain - see Z transform - properties of
convolution in the time domain property of the Laplace transform - see Laplace transform - properties of
convolution integral defined 6-8
  graphical evaluation of 6-8
convolution property of the Fourier transform - see Fourier transform properties of
Cooley and Tukey 10-18
cosine function - Fourier transform of see Fourier transform of common functions
$\cos\omega_0 t\ u_0(t)$ function - Fourier transform of see Fourier transform of common functions
Cramer's rule D-17

D

d2c MATLAB function 9-47
data points in MATLAB A-14
decade - definition of 11-12
decimation in frequency - see FFT algorithm
decimation in time - see FFT algorithm
deconv MATLAB function A-6
default color in MATLAB A-15
default line in MATLAB A-15
default marker in MATLAB A-15
default values in MATLAB A-12
delta (impulse) function
  definition of 1-11
  doublet 1-14
  Fourier transform of - see Fourier transform of common functions
  higher order 1-14
  nth-order 1-14
  sampling property of 1-12
  sifting property of 1-13
  triplet 1-14
demo in MATLAB A-2
DeMoivre's theorem 11-15
derivation of the Fourier transform from the Laplace Transform 8-25
determinant of a matrix - see matrix
determinant of order 2 - see matrix
DFT - common properties of
  even time function 10-9
  even frequency function 10-9
  frequency convolution 10-13
  frequency shift 10-12
  linearity 10-10
  odd time function 10-9
  odd frequency function 10-9
  time convolution 10-12
  time shift 10-11
DFT - definition of 10-1
  N-point 10-2, 10-16
diagonal elements of a matrix - see matrix
diagonal matrix - see matrix

- differentiation in complex frequency domain property of the Laplace transform - see Laplace transform - properties of
- differentiation in time domain property of the Laplace transform - see Laplace transform - properties of
- differentiation property of the Fourier transform 8-12 - see Fourier transform - properties of
- digital filter 11-1, 11-51, 11-70
  - Finite Impulse Response (FIR) 11-52
  - FIR 11-52
  - IIR 11-51
  - Infinite Impulse Response (IIR) 11-51
  - realization of
    - Direct Form **I** 11-70
    - Direct Form **II** 11-71
    - cascade (series) form 11-73
    - non-recursive 11-52
    - parallel form 11-70
    - recursive 11-52
    - series (cascade) form 11-73
- **Digital Filter Design** Simulink block 11-78
- digital filter design with Simulink 11-70
- **dimpulse** MATLAB function 9-29
- **Dirac** MATLAB function 1-20
- Direct Form I realization - see digital filter realization of
- Direct Form II realization - see digital filter realization of
- direct term in MATLAB 3-4
- discontinuous function - definition of 1-2
- Discrete Fourier Transform (DFT) 10-1
- discrete impulse response 9-40
- discrete-time system transfer function 9-40
- discrete unit step function 9-3
- discrete-time exponential sequence 9-16
- discrete-time systems 9-1
- discrete-time unit ramp function 9-18
- discrete-time unit step function 9-14
- **Display** block in Simulink B-18
- display formats in MATLAB A-31
- distinct eigenvalues - see eigenvalues
- distinct poles - see poles
- division of complex numbers C-4
- dot operator in MATLAB
  - division with A-21
  - exponentiation with A-21
  - multiplication with A-20
- double-memory technique see FFT Algorithm
- doublet - see delta function

E

- Editor Window in MATLAB A-1
- Editor/Debugger in MATLAB A-1
- **eig(x)** MATLAB function 5-17
- eigenvalues
  - distinct 15-1
  - multiple (repeated) 5-15
- eigenvector 5-19
- $e^{-j\omega t}$ $u_0(t)$ Fourier transform of - see Fourier transform of common functions
- element-by-element operation in MATLAB
  - division A-21
  - exponentiation A-21
  - multiplication A-20
- elements of the matrix - see matrix
- **ellip** MATLAB function 11-40
- elliptic filter - see filter
- elliptic filter design - see filter design
- **eps** in MATLAB A-22
- Euler's identities C-5
- even functions 6-4, 7-33
- even symmetry - see Fourier series - symmetry
- Excel's Analysis ToolPak 10-5
- **exit** MATLAB command A-2
- **expand** MATLAB symbolic function 3-10
- exponential form of complex numbers C-5
- exponential form of the Fourier series - see Fourier series
- exponential order function - definition of 2-2
- **eye(n)** MATLAB function D-7
- **eye(size(A))** MATLAB function D-7

F

- **factor(s)** MATLAB symbolic function 3-4
- Fast Fourier Transform (FFT) 10-1, 10-17
- **FDA Tool Digital Filter Design** Simulink block 11-82
- FFT algorithm
  - Category I 10-19
  - Category II 10-20
  - decimation in frequency 10-20
  - decimation in time 10-20
  - double-memory technique 10-20
  - in-place 10-20
  - natural input-output 10-20
- FFT definition of 10-1, 10-17
- **fft(x)** MATLAB function 10-5, 11-68
- Figure Window in MATLAB A-13
- filter - see also digital filter
  - active
    - high-pass 4-22, 4-32
    - low-pass 4-22, 4-32, 4-23, 4-35
  - all-pass 11-1, 11-94
  - all-pole 11-21
  - band-elimination 11-1, 11-8
  - band-pass 11-1, 11-7
  - band-stop - see band-elimination
  - Bessel 11-95
  - Chebyshev 11-10
    - Inverted 11-38
    - magnitude-square function 11-26
    - prototype 11-10
    - Type **I** 11-25
    - Type **II** 11-38
  - elliptic 11-39
  - high-pass 4-22, 4-30, 11-1, 11-4
  - low-pass 4-22, 4-30, 11-1, 11-2
  - low-pass analog filter
    - prototypes 11-10
  - maximally flat 11-14 (footnote)
  - notch (band-elimination) 11-8
  - phase shift 11-1, 11-94
  - RC high-pass 11-4
  - RC low-pass 11-2
  - RLC band-elimination 4-22, 4-31, 11-8
  - RLC band-pass 4-22, 4-31, 11-7
- filter design - see also digital filter
  - band-elimination 11-41
  - band-pass 11-41
  - Butterworth analog low-pass 11-14
  - Chebyshev
    - Type **I** 11-25
    - Type **II** 11-38
  - elliptic 11-39
  - high-pass 11-41
  - low-pass 11-14
- **filter** MATLAB function 11-63
- final value theorem in Z transform see Z transform - properties of
- final value theorem in Laplace transform see Laplace transform - properties of
- **find** MATLAB function 11-68
- Finite Impulse Response (FIR) digital filter see digital filter
- FIR - see digital filter
- first harmonic - see Fourier series harmonics of
- first-order circuit 5-1
- first-order simultaneous differential equations 5-1
- **Flip Block** command in Simulink B-11
- format in MATLAB A-31
- **fourier** MATLAB command 8-33
- Fourier series
  - exponential form of 7-31
  - method used in window functions E-17
  - trigonometric form of 7-2, 7-10
    - alternate form of 7-25
- Fourier series coefficients - evaluation of
  - numerical evaluation using Excel 7-46
  - numerical evaluation using MATLAB® 7-47
- Fourier series of common waveforms
  - full-wave rectifier 7-20, 7-24
  - half-wave rectifier 7-17, 7-20
  - square waveform
    - with even symmetry 7-9, 7-13
    - with odd symmetry 7-8, 7-12
  - sawtooth 7-9, 7-15
  - triangular 7-9, 7-16
- Fourier series - harmonics of
  - first 7-1, 7-10
  - second 7-1, 7-10
  - third 7-1, 7-10
- Fourier series - symmetry
  - even 7-7
  - half-wave 7-7, 7-34
  - odd 7-7
  - quarter-wave 7-7 (footnote)
  - types of 7-7
- Fourier integral - see Fourier transform
- Fourier transform
  - definition of 8-1
  - inverse of 8-1
  - special forms of 8-2

Fourier transform - properties of
  area under f(t) 8-15
  area under F(ω) 8-15
  conjugate time and frequency functions 8-13
  constant function 8-18
  frequency convolution 8-15
  frequency differentiation 8-13
  frequency shifting 8-11
  imaginary time functions - Fourier transform of 8-6
  linearity 8-9
  Parseval's theorem 8-16
  real time functions - Fourier transform of 8-3
  symmetry 8-9
  time convolution 8-14
  time differentiation 8-12
  time integration 8-13
  time scaling 8-10
  time shifting 8-11
Fourier Transform derivation from Laplace transform 8-25
Fourier transform of common functions
  $\cos\omega_0 t$ 8-19
  $\cos\omega_0 t \, u_0(t)$ 8-24
  delta ($\delta(t)$ and $\delta(t-a)$) 8-18
  $e^{-j\omega 0t}$ 8-19
  $e^{-j\omega 0t} u_0(t)$ 8-24
  signum (sgn(t)) 8-20
  $\sin\omega_0 t$ 8-20
  $\sin\omega_0 t \, u_0(t)$ 8-25
  unit step ($u_0(t)$) 8-22
Fourier transform of common waveforms
  combined rectangular pulses 8-29
  cosine within a rectangular pulse 8-30
  shifted rectangular pulse 8-28
  symmetrical rectangular pulse 8-27
  periodic time functions 8-31, 8-32
fourth-order all-pole low-pass filter
  see filter, low-pass
fplot in MATLAB A-27
frequency convolution in DFT
  see DFT - common properties of
frequency convolution in Fourier transform of - see Fourier transform - properties of
frequency differentiation in Fourier transform of - see Fourier transform - properties of
frequency shift in DFT
  see DFT - common properties of
frequency shift in Fourier transform
  see Fourier transform - properties of
frequency shift in Laplace transform
  see Laplace transform - properties of
freqz MATLAB function 11-57
full rectification waveform - Laplace transform of - see Laplace transform of common waveforms
full-wave rectifier - Fourier series of - see Fourier series of common waveforms
Function Block Parameters in Simulink B-10
function files in MATLAB A-26
fundamental frequency 7-1
fzero MATLAB function A-26

G

Gain block in Simulink B-9, B-18
gamma function 2-15
Gaussian elimination method D-19
generalized factorial function 2-15
Gibbs phenomenon 7-24
grid MATLAB command A-12
gtext MATLAB command A-13

H

half-wave rectifier - Fourier series of - see Fourier series of common waveforms
half-wave symmetry
  see Fourier series - symmetry
Heavyside MATLAB function 1-14
help in MATLAB A-2
Hermitian matrix - see matrix
higher order delta functions - see delta function
high-pass filter - see filter
high-pass filter design - see filter design

I

identity matrix - see matrix
ifft(x) MATLAB function 10-5
ifourier MATLAB function 8-33
IIR - see digital filter
ilaplace MATLAB function 3-4
imag(z) MATLAB function A-23
imaginary axis - definition of C-2
imaginary number - definition of C-2
imaginary time functions 8-6
  see Fourier transform - properties of
impedance
  capacitive 4-2
  inductive 4-2
  complex input 4-8, 4-9
improper integral - definition of 2-15
improper rational function - definition of 3-1, 3-13
impulse function - see delta function
impulse invariant method - see transformation methods for mapping analog prototype filters to digital filters
increments between points in MATLAB A-14
inductive admittance - see admittance
inductive impedance - see impedance
infinite impulse response - see digital filter
initial value theorem in Z transform
  see Z transform - properties of
initial value theorem in Laplace transform
  see Laplace transform - properties of
in-place FFT algorithm 10-20
  see FFT algorithm
integration in frequency in Laplace transform
  see Laplace transform - properties of
integration in time in Laplace transform
  see Laplace transform - properties of
Inverse Fourier transform
  see Fourier transform
Inverse Laplace transform 2-1
  see Laplace transform
inverse of a matrix - see matrix
Inverse Z transform - see Z transform
inversion integral 9-32
Inverted Chebyshev filter - see filter

J

j operator C-1

L

L' Hôpital's rule 2-16
laplace MATLAB function 2-27
Laplace integral - see Laplace transform
Laplace transform
  definition of 2-1
  Inverse of 2-1, 3-1
Laplace transform - properties of
  convolution in the complex frequency domain 2-12
  convolution in the time domain 2-11
  differentiation in complex frequency domain 2-6
  differentiation in time domain 2-4
  final value theorem 2-10
  frequency shift 2-4
  initial value theorem 2-9
  integration in
    complex frequency domain 2-8
    time domain 2-6
  linearity 2-3
  scaling 2-4
  time periodicity 2-8
  time shift 2-3
Laplace transform of common waveforms
  full-rectified 2-32
  half-rectified 2-26
  linear segment 2-23
  pulse 2-23
  rectangular periodic 2-25
  sawtooth 2-32
  triangular 2-24
Laplace transform of common functions
    transform of $e^{-at} \cos\omega t \, u_0(t)$ 2-22
    transform of $e^{-at} \sin\omega t \, u_0(t)$ 2-21
    transform of $e^{-at} u_0(t)$ 2-19
    transform of $\cos\omega t \, u_0(t)$ 2-20
    transform of $\delta(t)$ 2-18
    transform of $\delta(t-a)$ 2-18
    transform of $\sin\omega t \, u_0(t)$ 2-20
    transform of $t^n u_0(t)$ function 2-15
    transform of $t^n e^{-at} u_0(t)$ 2-19
    transform of $u_0(t)$ 2-14
    transform of $u_1(t)$ 2-14
leakage 10-13
left shift in discrete-time domain
  see Z transform - properties of
Leibnitz's rule 6
lims= MATLAB command A-27
line spectra 7-36
linear difference equation 9-38

linearity in DFT
  see DFT - common properties of
linearity in discrete-time domain
  see Z transform - properties of
linearity property in Fourier transform
  see Fourier transform - properties of
linearity property in Laplace transform
  see Laplace transform - properties of
linspace MATLAB command A-14
ln (natural log) A-13
log in MATLAB A-13
log(x) MATLAB function A-13
log10(x) MATLAB function A-13
log2(x) MATLAB function A-13
loglog MATLAB command A-13
long division of polynomials 9-36
lower triangular matrix - see matrix
low-pass analog filter prototypes - see filter
low-pass filter - see filter
lp2bp MATLAB function 11-43
lp2bs MATLAB function 11-43
lp2hp MATLAB function 11-43
lp2lp MATLAB function 11-43

M

magnitude-squared function 11-10
main diagonal of a matrix - see matrix
Math Operations in Simulink B-10
MATLAB Demos A-2
MATLAB's Editor/Debugger A-1
matrix (matrices)
  adjoint of D-21
  cofactor of D-13
  conformable for addition D-2
  conformable for subtraction D-2
  conformable for multiplication D-4
  congugate of D-8
  definition of D-1
  determinant D-10
    minor of D-13
    non-singular D-21
    singular D-21
  diagonal D-2, D-6
  diagonal elements of D-2
  elements of D-1
  Hermitian D-9
  identity D-7
  inverse of D-22
  left division in MATLAB D-25
  multiplication in MATLAB A-18
  power series of 5-9
  scalar D-7
  size of D-7
  skew-Hermitian D-9
  skew-symmetric D-9
  square D-1
  symmetric D-8
  trace of D-2
  transpose of D-8
  triangular
    lower D-6
    upper D-6
  zero D-2
matrix left division in MATLAB - see matrix
matrix multiplication in MATLAB - see matrix
matrix power series - see matrix
maximally flat filter - see filter
mesh(x,y,z) MATLAB function A-17
meshgrid(x,y) MATLAB command A-17
m-file in MATLAB A-2, A-26
minor of determinant - see matrix
MINVERSE Excel function D-27
MMULT Excel function D-27
modulated signals 8-12
multiple eigenvalues - see eigenvalues
multiple poles - see poles
multiplication by an in discrete-time domain
  see Z transform - properties of
multiplication by e^{-naT} in discrete-time
  domain - see Z transform - properties of
multiplication by **n** in discrete-time domain
  see Z transform - properties of
multiplication by $\mathbf{n}^2$ indiscrete-time domain
  see Z transform - properties of
multiplication of complex numbers C-3

N

NaN in MATLAB A-26
natural input-output FFT algorithm
  see FFT algorithm
network transformation
  resistive 4-1
  capacitive 4-1
  inductive 4-1
non-recursive realization digital filter
  see digital filter
non-singular determinant - see matrix
normalized cutoff frequency 11-15
notch filter - see filter
N-point DFT - see DFT - definition of
nth-order delta function - see delta function
numerical evaluation of Fourier coefficients
  see Fourier series coefficients
Nyquist frequency 10-13

O

octave defined 11-12
odd functions 6-5, 7-34
odd symmetry - see Fourier
  series - symmetry
orthogonal functions 7-2
orthogonal vectors 5-19
orthonormal basis 5-19

P

parallel form realization - see digital filter
Parseval's theorem - see
  Fourier transform - properties of
partial fraction expansion 3-1, 3-2, 9-25
  alternate method of 3-15
  method of clearing the fractions 3-15
phase angle 11-2
phase shift filter - see filter
picket-fence effect 10-14
plot MATLAB command A-10
polar form of complex numbers C-6
polar plot in MATLAB A-24
polar(theta,r) MATLAB function A-23
poles 3-1
  complex 3-5
  distinct 3-2
  multiple (repeated) 3-8
poly MATLAB function A-4
polyder MATLAB function A-7
polynomial construction from
  known roots in MATLAB A-4
polyval MATLAB function A-6
pre-sampling filter 10-13
pre-warping 11-55
proper rational function -
  definition of 3-1, 11-11
properties of the DFT
  see DFT - common properties of
properties of the Fourier Transform
  see Fourier transform - properties of
properties of the Laplace Transform
  see Laplace transform - properties of
properties of the Z Transform
  see Z transform - properties of

Q

quarter-wave symmetry 7-7 (footnote)
quit MATLAB command A-2

R

radius of absolute convergence 9-3
ramp function 1-9
randn MATLAB function 11-68
Random Source Simulink block 11-80
rationalization of the quotient C-4
RC high-pass filter - see filter
RC low-pass filter - see filter
real axis C-2
real number C-2
real(z) MATLAB function A-23
rectangular form C-5
rectangular pulse expressed in terms
  of the unit step function 1-4
recursive realization digital filter
  see digital filter
region of
  convergence 9-3
  divergence 9-3
relationship between state equations
  and Laplace Transform 5-30
residue 3-2, 9-41
residue MATLAB function 3-3, 3-12
residue theorem 9-20, 9-21
right shift in the discrete-time domain
  see Z transform - properties of
RLC band-elimination filter - see filter
RLC band-pass filter - see filter
roots of polynomials in MATLAB A-3
roots(p) MATLAB function 3-6, A-3
round(n) MATLAB function A-24
row vector in MATLAB A-3
Runge-Kutta method 5-1
running Simulink B-7

S

sampling property of the delta function
  see delta function
sampling theorem 10-13
sawtooth waveform - see Laplace
  transform of common waveforms
sawtooth waveform - Fourier series of
  see Fourier series of
    common waveforms
scalar matrix - see matrix
scaling property of the Laplace transform
  see Laplace transform - properties of
Scope block in Simulink B-12
script file in MATLAB A-2, A-26
second harmonic - see Fourier series
  harmonics of
semicolons in MATLAB A-8
semilogx MATLAB command A-12
semilogy MATLAB command A-12
series form realization - see digital filter
Shannon's sampling theorem
  see sampling theorem
shift of f[n] u_0[n] in discrete-time domain
  see Z transform - properties of
sifting property of the delta function
  see delta function
signal flow graph 10-23
signals described in math form 1-1
signum function - see Fourier transform
  of common functions
simout To Workspace block
  in Simulink B-12
simple MATLAB symbolic function 3-7
Simulation drop menu in Simulink B-12
simulation start icon in Simulink B-12
Simulink icon B-7
Simulink Library Browser B-8
sine function - Fourier transform of
  see Fourier transform of
    common functions
singular determinant - see matrix
Sinks library in Simulink B-18
sinω_0t u_0(t) Fourier transform of - see
  Fourier transform of common functions
size of a matrix - see matrix
skew-Hermitian matrix - see matrix
skew-symmetric matrix - see matrix
special forms of the Fourier transform
  see Fourier transform
spectrum analyzer 7-36
square matrix - see matrix
square waveform with even symmetry - see
  Fourier series of common waveforms
square waveform with odd symmetry - see
  Fourier series of common waveforms
ss2tf MATLAB function 5-33
stability 11-13
start simulation in Simulink B-12
state equations
  for continuous-time systems 5-1
  for discrete-time systems 9-45
state transition matrix 5-9
state variables
  for continuous-time systems 5-1
  for discrete-time systems 9-45
State-Space block in Simulink B-12
state-space equations
  for continuous-time systems 5-1
  for discrete-time systems 9-45
step function - see unit step function
step invariant method - see trans-
  formation methods for mapping analog
    prototype filters to digital filters
stop-band filter - see filter
string in MATLAB A-16
subplots in MATLAB A-18
summation in the discrete-time Domain
  see Z transform - properties of
symmetric matrix - see matrix
symmetric rectangular pulse expressed
  as sum of unit step functions 1-6
symmetric triangular waveform expressed
  as sum of unit step functions 1-6
symmetry - see Fourier series - symmetry
symmetry property of the Fourier transform
  see Fourier transform - properties of
system function - definition of 8-35

T

Taylor series 5-1
text MATLAB command A-14
tf2ss MATLAB function 5-33
theorems of the DFT 10-10
theorems of the Fourier Transform 8-9
theorems of the Laplace transform 2-2
theorems of the Z Transform 9-3
third harmonic - see Fourier
  series - harmonics of
time convolution in DFT
  see DFT - common properties of
time integration property of the Fourier
  transform - see Fourier
    transform - properties of
time periodicity property of the Laplace
  transform 2-8 - see Laplace
    transform - properties of
time scaling property of the Fourier
  transform - see Fourier
    transform - properties of
time shift in DFT
  see DFT - common properties of
time shift property of the Fourier transform
  see Fourier transform - properties of
time shift property of the Laplace transform
  see Laplace transform - properties of
title('string') MATLAB command A-12
trace of a matrix - see matrix
Transfer Fcn block in Simulink 4-17
Transfer Fcn Direct Form II
  Simulink block 11-71
transfer function of
  continuous-time systems 4-13
  discrete-time systems 9-38
transformation between
  s and z domains 9-22
transformation methods for mapping
  analog prototype filters to digital filters
    Impulse Invariant Method 11-52
    Step Invariant Method 11-52
    Bilinear transformation 11-53
transpose of a matrix - see matrix
Tree Pane in Simulink B-7
triangular waveform expressed in terms
  of the unit step function 1-4
triplet - see delta function
Tukey - see Cooley and Tukey

U

unit eigenvectors 5-19
unit impulse function (δ(t)) 1-8
unit ramp function (u_1(t)) 1-8
unit step function (u_0(t)) 1-2
upper triangular matrix - see matrix
using MATLAB for finding the Laplace
  transforms of time functions 2-27
using MATLAB for finding the Fourier
  transforms of time function 8-33

V

Vandermonde matrix 10-18
Vector Scope Simulink block 11-84

W

warping 11-54
window functions
  Blackman E-12
  Fourier series method for approximating
    an FIR amplitude response E-17
  Hamming E-9, E-31
  Hanning E-7, E-27
  Kaiser E-14, E-35
  other used as MATLAB functions E-15
  rectangular E-2
  triangular E-5, E-23
Window Visualization Tool in MATLAB E-4

X

xlabel MATLAB command A-12

Y

ylabel MATLAB command A-12

Z

Z transform
  computation of with contour
    integration 9-20
  definition of 9-1
  Inverse of 9-1, 9-25
Z transform - properties of
  convolution in the discrete
    frequency domain 9-9
  convolution in the discrete
    time domain 9-8
  final value theorem 9-10
  initial value theorem 9-9
  left shift 9-5
  linearity 9-3

    multiplication by a^n 9-6
    multiplication by e^{-naT} 9-6
    multiplication by **n** 9-6
    multiplication by $\mathbf{n}^2$ 9-6
    right shift 9-4
    shift of f[n] u_0[n] 9-3
    summation 9-7
Z Transform of discrete-time functions
    cosine function cosnaT 9-16
    exponential sequence e^{-naT} u_0[n] 9-16, 9-21
    geometric sequence a^n 9-11
    sine function sinnaT 9-16
    unit ramp function nu_0[n] 9-18, 9-21
    unit step function u_0[n] 9-14, 9-20
zero matrix - see matrix
zeros 3-1, 3-2
zp2tf MATLAB function 11-17